URX85x, MxL256xx

AnyWAN™ Broadband SoCs

AnyWAN™ SoC URX851
AnyWAN™ SoC URX850
AnyWAN™ SoC MxL25641

Data Sheet

MaxLinear Confidential

Revision 1.9, 2023-09-28
Reference ID 617102

Legal Notice

The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by MaxLinear, Inc. MaxLinear, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this document. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced into, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of MaxLinear, Inc.

EXCEPT AS OTHERWISE PROVIDED EXPRESSLY IN WRITING BY MAXLINEAR, AND TO THE MAXIMUM EXTENT PERMITTED BY LAW: (A) THE MAXLINEAR PRODUCTS ARE PROVIDED ON AN “AS IS” BASIS WITHOUT REPRESENTATIONS OR WARRANTIES OF ANY KIND, INCLUDING WITHOUT LIMITATION ANY IMPLIED OR STATUTORY WARRANTIES AND ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; AND (B) MAXLINEAR DOES NOT GUARANTEE THAT THE PRODUCTS WILL BE FREE OF ERRORS OR DEFECTS. MAXLINEAR PRODUCTS SHOULD NOT BE USED IN ANY EMERGENCY, SECURITY, MILITARY, LIFE-SAVING, OR OTHER CRITICAL USE CASE WHERE A FAILURE OR MALFUNCTION COULD CAUSE PERSONAL INJURY OR DEATH, OR DAMAGE TO OR LOSS OF PROPERTY. USERS ASSUME ALL RISK FOR USING THE MAXLINEAR PRODUCTS IN SUCH USE CASE. CUSTOMERS AND USERS ARE SOLELY RESPONSIBLE FOR USING THEIR OWN SKILL AND JUDGMENT TO DETERMINE WHETHER MAXLINEAR PRODUCTS ARE SUITABLE FOR THE INTENDED USE CASE.

MaxLinear, Inc. may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from MaxLinear, Inc., the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

MaxLinear, the MaxLinear logo, any other MaxLinear trademarks (including but not limited to MxL, Full-Spectrum Capture, FSC, AirPHY, Puma, AnyWAN, VectorBoost, MXL WARE, and Panther), and the MaxLinear logo on the products sold are all property of MaxLinear, Inc. or one of MaxLinear's subsidiaries in the U.S.A. and other countries. All rights reserved.

All third-party products, company names and logos are trademarks™ or registered® trademarks and remain the property of their respective holders/owners. Use of such marks does not imply any affiliation with, sponsorship or endorsement by the owners/holders of such trademarks. All references by MaxLinear to third party trademarks are intended to constitute nominative fair use under applicable trademark laws.

The URLs provided are for informational purposes only; they do not constitute an endorsement or an approval by MaxLinear of any of the products or services of the corporation or organization or individual. MaxLinear bears no responsibility for the accuracy, legality or content of the external site or for that of subsequent links. Contact the external site for answers to questions regarding its content.

MaxLinear, Inc.
5966 La Place Court, Suite 100
Carlsbad, CA 92008
Tel.: +1 (760) 692-0711
Fax: +1 (760) 444-8598
www.maxlinear.com

Revision History

Current:	**Revision 1.9, 2023-09-28**
Previous:	**Revision 1.8, 2023-09-11**
Page	**Major changes since previous revision**
3-4	No changes to technical content. This revision only provides further details on the changes in Rev. 1.8.
Page	**Major changes introduced with revision 1.8**
All	Removed the Preliminary statement.
All	Updated mentions of USB 3.1 to USB 3.2. See **Table 3**.
24	Updated the **Preface** to add the MxL31712 device.
24	Updated **Figure 1 Example Configuration of Tri-band Wi-Fi 7 Multi-WAN Home Router Based on URX851** to show the Wi-Fi 7 configuration.
25	Updated **Section 1.1 High Level Architecture Overview** to specify that DDR content protection is only available in URX850 by default.
29	Added **Table 3 USB 3.1 to USB 3.2 Interface Speed Mapping**.
30	Updated **Table 4 Feature Overview - Interface Modules**. • **HSIO Combo Subsystem Interface/Ethernet/SATA SerDes Interface Module #1** – Spec Compliance • **HSIO Combo Subsystem Interface/Ethernet/SATA SerDes Interface Module #2** – Spec Compliance • **Flexible Processing Engine** – No. of Instances • **QoS Engine (QoS) Hardware** – No. of Queues – No. of Schedulers • **Wireless Mode** - Applicable only for URX851/URX850 • Footnote [4)]
43	Updated the 1000BASE-KX/SGMII to 10G-KR/10G/5G/2.5G-USXGMII entry in **Table 5 XFI Ethernet Modes (Applicable Only for URX851/URX850/MxL25641)**.
47	Updated **Table 9 GPIO Interface Multiplexing** to remove the ZSI interface details.
57	Updated **Section 2.3.11 USB Port 0** to align with USB 3.2.
58	Updated **Section 2.3.12 USB Port 1** to align with USB 3.2.
59	Updated **Table 22 GPIO/Flash/I²C/I2S/SPI/PCM Signals** to remove the ZSI interface details.
89	Updated **Section 2.4 Digital Signal Ball (Pin) Reset Property and Drive Strength** to add references to **Figure 15 Reset System**.
92	Updated **Table 42 External Signals** to align with USB 3.2.
95	Updated **Table 45 GPIO Interface Multiplexing** to remove the ZSI interface details.
99	Updated ball names in **Table 47 PCIe 4.0 Interface #11 Signals**.
100	Updated ball names in **Table 49 PCIe 4.0 Interface #21 Signals**.
102	Updated **Table 52 USB1 Signals**.
103	Updated **Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals** to remove the ZSI interface details.
133	Updated **Section 3.4 Digital Signal Ball (Pin) Reset Property and Drive Strength** to add references to **Figure 15 Reset System**.

Revision History

Current:	**Revision 1.9, 2023-09-28**
Previous:	**Revision 1.8, 2023-09-11**
139	Updated **Section 4.1.2.1 Intel Atom Processor Core** to remove mention of MaxLinear secure key.
149	Updated the future proven architecture details in **Section 4.3.1 Feature List and Conceptual Ideas**.
152	Restructured information within **Section 4.3.2.1 NoC Access Control Security**.
183	Updated **Section 6.1.2 DDR4, DDR3, and LPDDR SDRAM Controller** to remove details on DDR4-3DS support.
237	Added **Application Notes for Isochronous Support** for isochronous devices connections.
243	Updated **Section 8.1 Packet Processing v4 (PPv4)** to reflect the number of microcontroller subsystems.
277	Updated FIFO details in **Section 9.1.1 ASC Features**.
278	Updated **Section 9.1.2 General Operation** to add a warning on the ASC 12.6MB baud-rate support.
289	Updated **Table 119 Port 0 Functions** to remove the ZSI interface details.
293	Updated **Table 120 Port 1,2,3,4 Functions** to remove the ZSI interface details.
306	Updated **Table 122 Port 0 Functions MxL25641** to remove the ZSI interface details.
311	Updated **Table 123 Port 1,2,3,4 Functions MxL25641** to remove the ZSI interface details.
334	Updated **Section 9.13 Voice Subsystem** with clearer system support features.
-	Removed these sections: • 9.13.1.2 Interrupts • 9.13.1.2.1 C55 Interrupt to CPU • 9.13.1.2.2 Interrupts to C55
339	Updated **Section 9.13.1.2 TDM and SPI** to remove details on the C55 interrupts.
339	Updated details on cached and uncached access in **Section 10.1.3.3 Vt-x2 SAI Control**.
341	Updated details on Power Management in **Section 10.2.1.1 Features**.
345	Updated **Section 10.3 DDR Content Protection with In-line Encryption Device Engine** to specify that DDR content protection is only available in URX850 by default.
372	Updated the Packet Processor Performance Class details in **Table 142 MxL25641 SKU Details**.
376	Updated parameters and values in **Table 146 Typical System Power Supply Current for URX851/URX850**.
380	Updated parameters and values in **Table 148 Typical System Power Supply Current for MxL25641**.
437	Updated **Terminology**.
440	Updated **Literature References**.

Third-Party Trademark and Registered Trademark Declarations

Aavid® (now Boyd)	Aavid is a registered trademark of Boyd Corporation.
AMBA®	Arm and AMBA are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
Android™	The Android name, the Android logo, and other Google trademarks found at https://about.google/brand-resource-center/trademark-list/ are property of Google LLC.
ANSI®	ANSI, the ANSI logo, and numerous other identifiers containing ANSI are registered trademarks, service marks, and accreditation marks of the American National Standards Institute (ANSI).
ARC®	ARC is a trademark of Synopsys, Inc. in the U.S. and/or other countries.
BENAND™	BENAND is a trademark of KIOXIA Corporation.
Cisco®	Cisco is a registered trademark or trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries. A list of other Cisco marks is available at https://www.cisco.com/c/en/us/about/legal/trademarks.html.
DOCSIS®	DOCSIS is a registered trademark of Cable Television Laboratories, Inc.
GigaDevice®	GigaDevise is a registered trademark of GigaDevice Semiconductor Inc.
IEEE®	IEEE, the IEEE logo, and other IEEE logos and titles are registered trademarks or service marks of The Institute of Electrical and Electronics Engineers, Incorporated.
Intel®	Intel and the Intel logo are trademarks of Intel Corporation or its subsidiaries.
Intel Atom®	Intel, the Intel logo, and Intel Atom are trademarks of Intel Corporation or its subsidiaries.
ITU-T®	ITU-T is a registered trademark of the ITU, a specialized agency of the United Nations.
JEDEC®	JEDEC is a registered trademark of the JEDEC Solid State Technology Association.
Linux®	Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.
Macronix®	Macronix, MXIC, MXIC logo, XtraROM, NBit and Macronix NBit are trademarks of Macronix International Co., Ltd.
Marvell®	Marvell and the Marvell logo are trademarks of Marvell or its affiliates. Visit https://www.marvell.com/company/trademarks.html for a list of Marvell trademarks.

Micron®	Micron® and the Micron orbit logo are trademarks of Micron Technology, Inc.
MIPS®	MIPS and the MIPS logo are trademarks or registered trademarks of MIPS Tech LLC in the United States and other countries.
Motorola®	MOTOROLA and the Stylized M Logo are trademarks or registered trademarks of Motorola Trademark Holdings, LLC.
NIST®	NIST is a registered trademark of the National Institute of Standards and Technology.
NXP®	NXP is a trademark of NXP B.V.
OpenWrt®	OpenWrt is a registered trademark owned by Software in the Public Interest, Inc.
PCI Express®	PCI-SIG®, PCIe®, and PCI Express® are registered trademarks and/or service marks of PCI-SIG.
PCIe®	PCI-SIG®, PCIE®, and PCI Express® are registered trademarks and/or service marks of PCI-SIG.
Philips®	The PHILIPS wordmark and shield emblem are registered trademarks of Koninklijke Philips N.V.
RDK®	RDK is a registered trademark of RDK Management, LLC.
Samsung®	Samsung is a registered trademark of Samsung Electronics Co., Ltd.
Spansion®	Spansion is a registered trademark of Infineon Technologies AG.
USB-C®	USB Type-C® and USB-C® are registered trademarks of USB Implementers Forum.
Yocto Project®	Yocto Project® is a trademark of The Linux Foundation in the United States and/or other countries.
Vishay®	Vishay is a registered trademark of Vishay Intertechnology, Inc.
Wi-Fi®	Wi-Fi®, the Wi-Fi logo, and Wi-Fi Alliance® are registered trademarks of Wi-Fi Alliance®.

All other trademarks or registered trademarks are the property of their respective holders/owners.

Table of Contents

List of Figures

List of Tables

Preface

This document describes the hardware specification of the AnyWAN™ SoC URX851/URX850/MxL25641 device. It includes an architectural overview, pin description, electrical characteristics, and mechanical characteristics, which provides sufficient details to design and build a solution around the AnyWAN™ SoC URX851/URX850/MxL25641.

The target audience for this document is primarily system architects, solution architects, software architects, and board designers planning to develop an application around the AnyWAN™ SoC URX851/URX850/MxL25641.

This document is also used by other system engineers such as software developers and system test engineers.

This document uses these synonyms to simplify matters:

URX85x

Synonym used for the AnyWAN™ Broadband SoCs URX85x, which includes the AnyWAN™ SoC URX851 and AnyWAN™ SoC URX850 devices

URX851

Synonym used for the AnyWAN™ SoC URX851

URX850

Synonym used for the AnyWAN™ SoC URX850

MxL256xx

Synonym used for the AnyWAN™ Broadband SoCs MxL256xx, which includes the AnyWAN™ SoC MxL25641

MxL25641

Synonym used for the AnyWAN™ SoC MxL25641

MxL31712

Synonym used for the Wi-Fi 7 SoC MxL31712

1 Overview

This chapter provides an overview of the AnyWAN™ SoC architecture and main features.

Product Overview

AnyWAN™ Broadband SoCs URX85x is a family of single-chip SoCs with 10 Gbps interfaces and embedded Intel Atom micro-architecture designed for the next generation of home routers and gateways and small office / home office gateway applications.

AnyWAN™ Broadband SoCs MxL256xx are the dual Intel Atom Core versions of this SoC class targeted for mid range home routers and gateways.

The SoC architecture is scalable across multiple WAN technologies – including fiber, cable, Ethernet, fixed wireless and DSL/G.fast - and a range of value and performance tiers. This provides the flexibility to use a common software framework to maximize investments across gateway designs.

Feature Highlight of URX851/URX850

- CPU
  - Four embedded 2.0 GHz Intel Atom CPU cores with 2 MB L2 cache
- Routing
  - 10 Gbps full duplex line rate with small (64 byte) packets
  - 35 Gbps and 30 million packets per second
- Interfaces
  - 10G PON MAC (applicable to URX851 only)
  - Four 2.5 Gbps Ethernet PHYs
  - Four dedicated PCIe 3.0
  - Four multiplexed XFI/PCIe 4.0/SATA
  - Dual-lane PCIe (combine 2x PCIe)
  - XFI interfaces configurable as USXGMII, RXAUI, SGMII+, SGMII, and K/KR
  - All high speed interfaces configurable as LAN or WAN
  - Two USB 3.2, 10 Gbps (Type-A or Type-C) including Type-C multiplexer circuitry
  - PCM and SPI interfaces for corded (FXS) and cordless (DECT) voice
  - Interface for far field voice/smart speakers and all common IoT interfaces
- Hardware Security
  - Network-on-chip architecture: configurable and firewalled
  - One-time programmable storage for platform root keys
  - Secure boot
  - Signature verification
  - True random number generation
  - Crypto accelerators
  - Secure storage (for example for key storage in flash memory)
  - IPsec end-point termination up to 10 Gbps line rate
- Scalability
  - Common BSP with reference software that builds with Yocto (RDK), prpl, and OpenWrt (UGW) build system
- Software Virtualization
  - 4x Intel Atom CPU
  - Hardware virtualization with VT-x2 and EPT
  - Virtual Machines
- Applications
  - Cable/DOCSIS modem and home gateway
  - Fiber optics Home Gateway Unit (HGU)
  - Ethernet Wi-Fi home router

  - DSL/G.fast home gateway
  - Fixed Wireless Access (FWA) home gateway
  - Multi-WAN gateway such as Telco home routers with integrated PON and DSL
  - Hybrid WAN gateways aggregating DSL and FWA
  - Other similar applications such as consumer NAS devices, DPUs, or small cells.
- Adaptive Power Management
  - Dynamic voltage and frequency scaling, clock gating, power domains, power gating

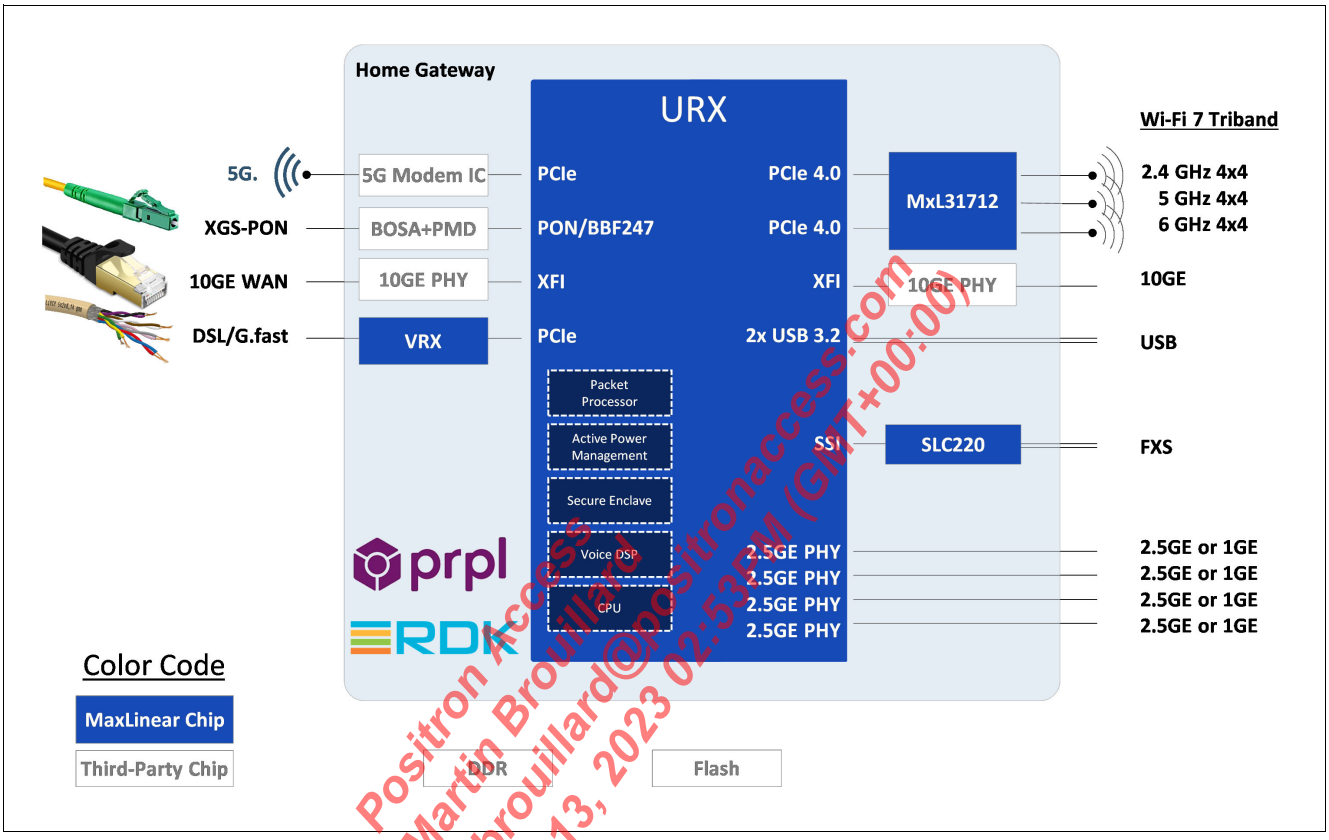


Figure 1 Example Configuration of Tri-band Wi-Fi 7 Multi-WAN Home Router Based on URX851

1.1 High Level Architecture Overview

The high level architecture overview includes the functional block diagram and a description of the building blocks.

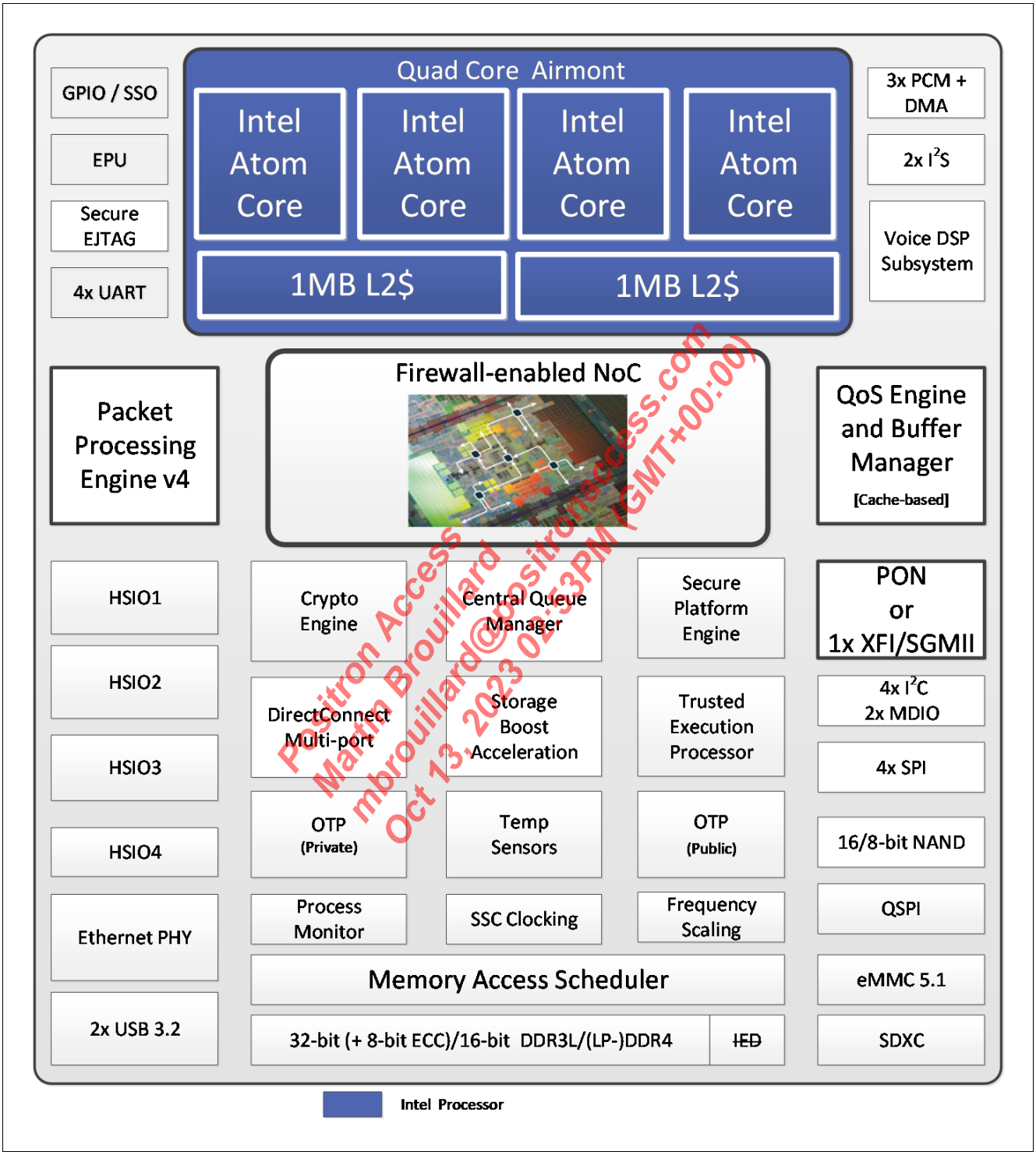


Figure 2 URX851 Block Diagram

Figure 2 shows the block diagram with the modules integrated for URX851.

The networking processing subsystem consists of four Intel Atom cores, based on Airmont architecture; 2 MB L2 cache is integrated in the subsystem.

The Network-on-Chip (NoC) is a packetized interconnect architecture which connects all SoC modules together, inside the NoC, a customized firewall defines the access rights for master modules to reach a specific peripheral or a memory region.

Packet Processing Engine V4 is the 4th generation hardware acceleration engine to handle packet bridging and routing with software defined session context. The Quality of Service (QoS) and Buffer Manager (BM) are part of PPv4 subsystem in design, the cached based BM manages buffer allocation and free for various data path interfaces, a large amount of pointers can be stored in DDR with cached entries on chip.

The BM receives traffic from enqueue ports and dispatches them into dequeue ports.

The Direct Connect Accelerator (DC), DirectConnect in **Figure 2**, is a MaxLinear-developed mechanism designed to offload traffic from PCIe DC-compliant connected peripherals, such as MaxLinear Wi-Fi Access Point (AP) devices, from the CPU.

These security modules are integrated:

- Trusted Execution Processor (TEP) manages security modules and functionality. It provides the interface to the non-secure world.
- Secure Platform Engine (SPE) provides accelerated crypto operation assistance to TEP.
- One Time Programmable (OTP) stores on chip security asserts.
- Secure EJTAG provides secure debug ON/OFF control.

In-line Encryption Device (IED) engine provides encryption/decryption operation towards the DDR, only available in URX850 by default.

The crypto engine provides networking data security packet processing for IPsec/TLS applications. MACsec provides link layer protection for Ethernet and EPON.

Power management features are built-in with the Energy Process Unit (EPU) as a central power management controller. Process monitors and temperature sensors provide run time inputs to EPU. The EPU does dynamic voltage and frequency scaling as well as gating.

Both High Speed IO (HSIO) and standard peripherals interfaces features are described in **Section 1.2**.

The URX850 is the product variant of URX85x without PON MAC and interface. **Figure 3** shows its block diagram.

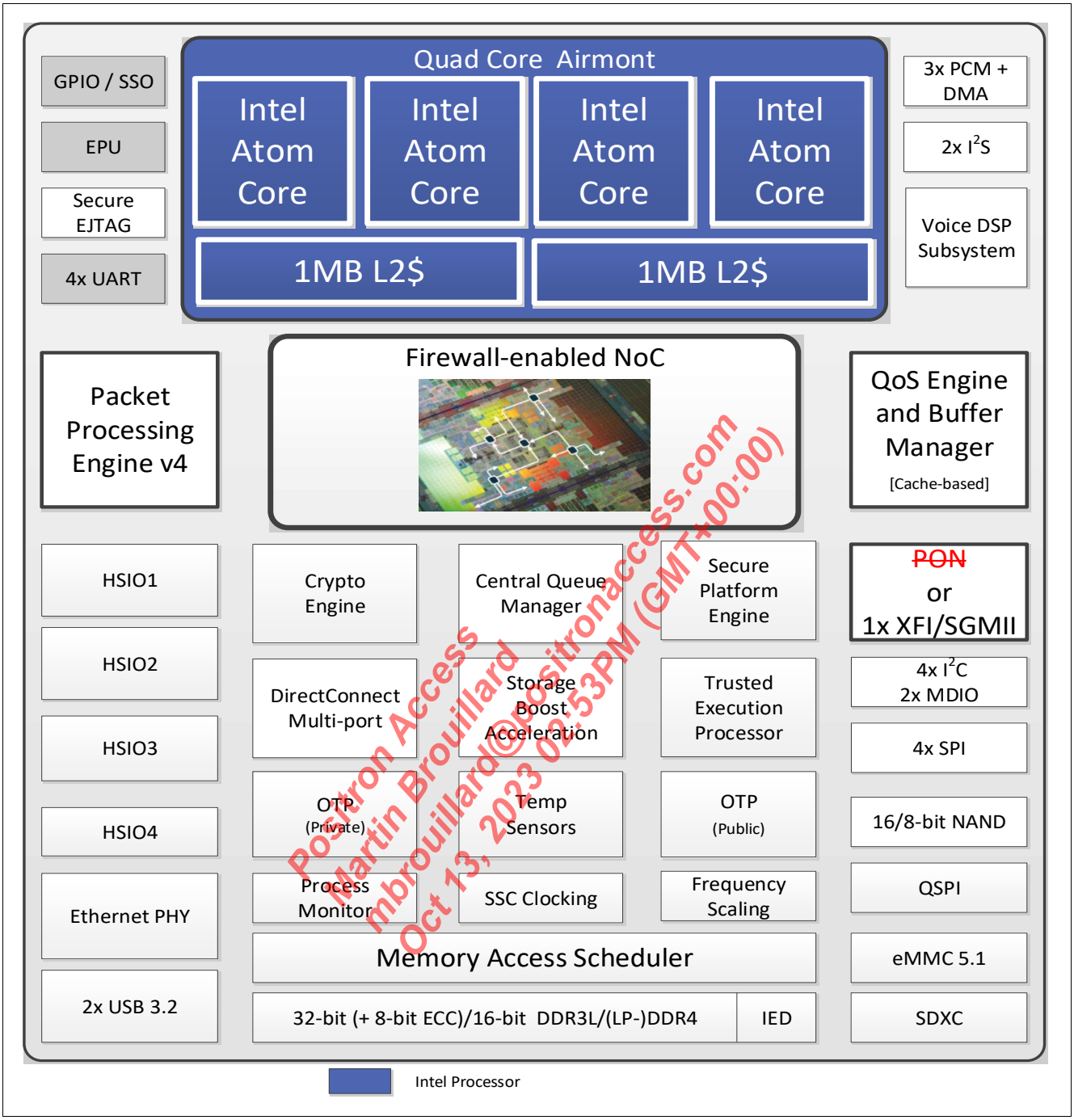


Figure 3 URX850 Block Diagram

Note: URX850 does not support PON.

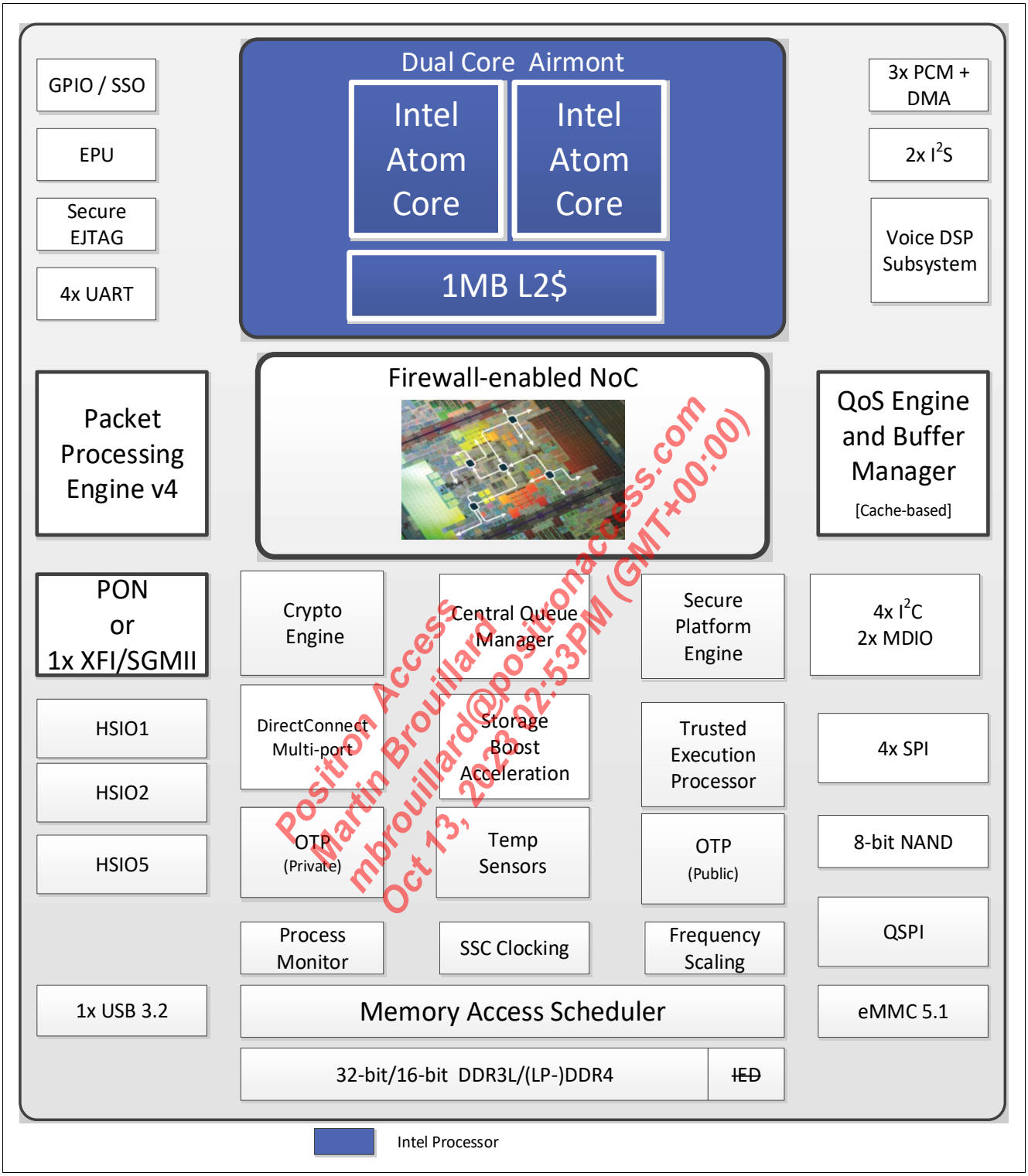


Figure 4 MxL25641 Block Diagram

1.1.1 Product Variants Information

This section gives the sales information for the different product variants.

Table 1 Product Sales Variants Information

Description / Package Type	App Core Intel Atom Clock (MHz)	HSIO (No. of lanes)	DDR (MHz) Security	ETH	Accelerated IPsec VPN	EMMC and SD Ports	Feature Credential Enabled[1)]
URX851B1BE/ EL V1.1-G / PG-FCBGA-837-C-1A /T&R	4x 2.0 GHz	8x + PON/XFI	Up to 933 +ECC	4x 2.5GE PHY	10 Gbps	EMMC5.1 + SD Card	Secure boot with protected regions[2)]
URX850B1BE EL V1.1-G / PG-FCBGA-837-C-1A /T&R	4x 2.0 GHz	8x + XFI	Up to 933 +ECC	4x 2.5GE PHY	10 Gbps	EMMC5.1 + SD Card	Secure boot with protected regions[2)]
MxL25641-AV-T EL V1.1-G / PG-FCBGA-577 /T&R	2x 1.7 GHz	4x + PON/XFI	Up to 800	USXGMII /XFI	5 Gbps	EMMC5.1	Secure boot with protected regions[2)]

1) The Feature Credential (FC) is required to enable these premium features.
2) This FC allows a secure boot to load an encrypted and authenticated product image.

Table 2 HSIO Interface Configuration Details

Interface	Availability
Product	**URX851/URX850**
HSIO1	2x XFI/SGMII, or two x1 PCIe 4.0, or 2x SATA3.2, or 1x RXAUI, or one x2 PCIe 3.0
HSIO2	2x XFI/SGMII, or two x1 PCIe 4.0, or 2x SATA3.2, or 1x RXAUI, or one x2 PCIe 3.0
HSIO3	Two x1 PCIe 3.0, or one x2 PCIe 3.0
HSIO4	Two x1 PCIe 3.0, or one x2 PCIe 3.0
2x USB 3.1	2x USB 3.2 Gen 1x1/Gen 2x1; Type-A or Type-C, backward compatible to USB 2.0/1.1
Product	**MxL25641**
HSIO1	2x PCIe 4.0, 2x XFI/SGMII
HSIO2	2x PCIe 4.0, 2x XFI/SGMII
HSIO5	1x USXGMII/XFI
1x USB 3.1	1x USB 3.2 Gen 1x1/Gen 2x1; Type-A, backward compatible to USB 2.0/1.1

Attention: The USB Implementers Forum updated the USB specification for the third generation USB from USB 3.1 to USB 3.2.

Table 3 shows the related interface name mapping.

Table 3 USB 3.1 to USB 3.2 Interface Speed Mapping

Speed	Name in USB 3.1	Name in USB 3.2
5 Gbps	Gen 1	Gen 1x1
10 Gbps	Gen 2	Gen 2x1

1.2 Interfaces

The chip provides these interfaces:

- User and network interfaces
- Processor interfaces
- Test interfaces

Table 4 Feature Overview - Interface Modules

Features	URX851/URX850	MxL25641
Main Processor		
Core Version	Intel Atom Airmont Core	
No. of Cores Total [No. of Cores per M:Module]	4 [2:2]	2 [2:1]
OS Type	64-bit	
CPU Frequency	624/936/1092/1248/1560/1716/ 1872/2028 MHz	624/936/1092/1248/1560/1716 MHz
Vectored Interrupts	Supported	
TLBs	Per core: • 48 instruction TLB • 16 entry, 4-way large page TLB • 256 entry, 4-way small page DTLB • 32 micro TLB	
Instruction Cache	32 kbyte, 8-way per core	
Data Cache	24 kbyte, 6-way per core	
No. of Outstanding Data Cache Misses	16 per core	
Outstanding Non-blocking Loads	16 per core	
Write-back and Write-through Support	Supported	
Virtual Address Range	> 64 GB	
Physical Address Range	16 GB DDR + 1 GB MMIO	
L2$	2 MB (2x1 MB per module)	1 MB (1 MB in one module)
External Bus Unit + NAND		
No. of Instances	1	
Data Bus	16/8-bit	8-bit
NAND Flash Mode	2 CS, 16/8-bit multiplexed address/data	2 CS, 8-bit multiplexed address/data
8-bit NAND Flash Mode Maximum	2 devices. Memory size unlimited from controller side, limited by flash device. ONFI mode, BENAND mode supported	
NAND Type	• Maximum of 8K page • Maximum of 5 address cycles • Up to 2 plane mode	
NAND Access Mode	• SLC hardware sequencer • MLC hardware sequencer	
EBU DMA Port	Half duplex DMA port	
Read Access WAIT State Control	0 - 7 EBU cycles	

Table 4 Feature Overview - Interface Modules (cont'd)

Features	URX851/URX850	MxL25641
ECC Location Inside Spare Area	Configurable to start or offset of 6 byte in spare area	
Customer Content Programming	32-bit customer programmable content added after ECC in spare area	
Erased Page Detection	Using bit masking pattern to mask the ECC code upon generation to ensure erased page ECC is all 0xFF. On decoder, applies same mask to unmask. Extra detection of erased pages is not required.	
HW ECC Algorithm Supporting	Based on BCH algorithm • 4-bit ECC per 528 bytes, per 1024 bytes • 8-bit ECC per 528 bytes, per 1024 bytes • 24 bit per 1024 bytes	
DDR Memory Controller		
No. of Instances	1	
Memory Type	• 16/32-bit DDR-3L • 16/32-bit DDR-4 • 32-bit DDR-4 with 8-bit hamming code ECC • 32-bit LPDDR4	• 16/32-bit DDR-3L • 16/32-bit DDR-4 • 32-bit LPDDR4
Chip Select 0	Up to 16 Gbytes	Up to 8 Gbytes
Chip Select 1		
Bank Select	2 (2 bank group select)	
Address lines	16	
On-die Termination (parallel) in Pad for DDR4/LPDDR4 Mode	34/60/120 Ω (programmable)	
On-die Termination [parallel] in Pad for DDR3L Mode	40/60/120 Ω (programmable)	
ZQ Calibration of Pull-up/Pull-down Input Impedance as well as Output Impedance of SSTL I/O	External 120 Ω ±1 %	
Supported Memory Devices with Standard Interface Clock	• Up to DDR3L-2133 • Up to – DDR4-3200 – LPDDR4-3200	
Speed Grade Memory Devices	LPDDR4-3733[1)]	Not supported
No. of Command Queues with Look Ahead Logic	DDR scheduler using threads to give priority access: the scheduler is configured with eight ports with traffic separation of initiator grouping.	
eMMC Interface		
Spec Compliance	eMMC v5.1	
No. of Instances	1	
SDXC Interface		
Spec Compliance	SD/SDIO 3.0	–
No. of Instances	1	–
USB Interfaces		
Spec Compliance	USB 3.2 Gen 1/USB 3.2 Gen 2 + USB 2.0/USB 1.1	

Table 4 Feature Overview - Interface Modules (cont'd)

Features	URX851/URX850	MxL25641
No. of Lanes	2 = Type-C multiplexer on-chip	1 = Type-A
No. of Instances	2	1
PCI Express 3.0 Interfaces[2)]		
No. of Lanes	4 independent single lanes or 2 dual lanes or 2 single lanes plus 1 dual lane	–
No. of Instances	4	–
Generate Reference Clock to End Point Outputs	4	–
PHY Data Rate	4x 8/5/2.5 GTs each	–
HSIO Combo Subsystem Interface/Ethernet/SATA SerDes Interface Module #1		
No. of PCIe Controller Instances	2	2
No. of SATA Controller Instances	2	0
No. of XFI/SFI Controller Instances	2	2
Spec Compliance	XFI/SGMII/USXGMII/RXAUI/ 10GBASE-KR PCIe 4.0 (16/8/5/2.5 GTs) SATA 3.2 (6/3/1.5 Gbps)	XFI/SGMII/USXGMII/10GBASE-KR PCIe 4.0 (16/8/5/2.5 GTs)
PCIe Inline Protection	• Bus tampering detection • AES-CTR crypto for TLP packets • Independent write/read path keys • Ratable key sets • Implemented for RC controller PCIe 10	
No. of Lanes	2 independent PCIe lanes or 2 independent XFI/SFI/USXGMII lanes or 1 RXAUI dual lane or 1 PCIe dual lane or 2 SATA lane	2 independent PCIe lanes or 2 independent XFI/SFI/USXGMII lanes or 1 PCIe dual lane
HSIO Combo Subsystem Interface/Ethernet/SATA SerDes Interface Module #2		
No. of PCIE Controller Instances	2	2
No. of SATA Controller Instances	2	0
No. of XFI/SFI Controller Instances	2	2
Spec Compliance	XFI/SGMII/USXGMII/RXAUI/ 10GBASE-KR PCIe 4.0 (16/8/5/2.5 GTs) SATA 3.2 (6/3/1.5 Gbps)	XFI/SGMII/USXGMII/RXAUI/ 10GBASE-KR PCIe 4.0 (16/8/5/2.5 GTs)
PCIe Inline Protection	• Bus tampering detection • AES-CTR crypto for TLP packets • Independent write/read path keys • Ratable key sets • Implemented for RC controller PCIe 20	

Table 4 Feature Overview - Interface Modules (cont'd)

Features	URX851/URX850	MxL25641
No. of Lanes	2 independent PCIe lanes or 2 independent XFI/SFI/USXGMII lanes or 1 RXAUI dual lane or 1 PCIe dual lane or 2 SATA lane	2 independent PCIe lanes or 2 independent XFI/SFI/USXGMII lanes or 1 PCIe dual lane
Ethernet Twisted Pair Interface		
Rate	2500/1000/100/10 Mbps	–
IEEE Standards	• 10BASE-T • 100BASE-TX • 1000BASE-T • 2.5GBASE-T	–
No. of 2.5 GE Instances	4	–
No. of 1 GE Instances	--	–
Common Clock Distribution Module	1	–
EEE Standard	Supported	–
MACsec Inline Engine		
No. of Instances	1 (selectable for HSIO lane10/11, HSIO lane 20/21, and PON XFI interface)	
MACsec Standards	• 802.1AE 128-bit and 256-bit • 802.1AEbw	
Security Channels	16	
Security Associations	32	
Packet Tagging	Supports 8B special tag and 10B additional packet information at the end of the packet	
MACsec Classification	Support MACsec classification based on special tag field and other packet header field	
Field Update	Supports packet length field update in special tag in encryption data path	
Engine Speed	10 Gbps bi-directional	
Memory Copy DMA Engine		
No. of Instances	3	
ToE DMA 0	ToE-DMA0	
No. of DMA Channels	32 [16:16] (16 Tx and 16 Rx channels)	
Memcpy Mode	Copy data between any two 36-bit memory locations	
Managed by	**Storage Application Acceleration Engine - TCP Offloading Engine (ToE)**	
ToE DMA 1	ToE-DMA1	
No. of DMA Channels	32 [16:16]	
Memcpy Mode	Copy data between any two 36-bit memory locations	
Managed by	**Storage Application Acceleration Engine - TCP Offloading Engine (ToE)**	
DMA-MCPY	DMA3-MCPY	
No. of DMA Channels	16 [8:8]	
Memcpy Mode	Copy data between any two 36-bit memory locations	

Table 4 Feature Overview - Interface Modules (cont'd)

Features	URX851/URX850	MxL25641
Managed by	Memory Copy DMA Acceleration	
Data Path DMA Engine		
No. of Instances	**4**	
Peripheral DMA Engine - DMA0		
No. of Instances	1	
Connected Modules on D+	• D+ based ports – NAND controller – SPI0 – SPI1 – DMA FCC1 – DMA FCC2 – SPI2 – SPI3 • Memcpy based ports – DMA FCC0 [2:0]	
Managed by	Intel Atom CPU	
Packet Processor v4 (PPv4) Hardware		
IPv4 Routing Entries (Acceleration Path)	16M	
IPv6 Routing Entries (Acceleration Path)	16M	
No. of Logical Ports	256	
No. of Sub-interface over Logical Ports	64K	
Flexible Processing Engine		
No. of Instances	4 ingress + 4 egress processors	
IPv4/IPv6 Entries	Up to 16 million route entries as tables in the DDR memory	
Session Search	Hardware-based hashing and search	
Central Queue Manager (CQM) Hardware		
No. of Instances	1	
No. of CPU/LRO Ports [Enqueue:Dequeue]	[16:19]	[13:13]
No. of DMA/TSO Ports [Enqueue:Dequeue]	[10:48]	[8:48]
PON Ports [Enqueue:Dequeue]	[--:64]	
DirectConnect Ports [Enqueue(RXOUT):Dequeue(TXIN)] [Buffer Return(TXOUT):Buffer Replenish(RXIN)]	[8:8] [8:8]	[4:4] [4:4]

Table 4 Feature Overview - Interface Modules (cont'd)

Features	URX851/URX850	MxL25641
VPN Adapter (HOST IF + ACA-DMA)		
Data Path Adaptation	Adapt the **Crypto Engine Hardware (VPN Accelerator)** to the DirectConnect ports on **Central Queue Manager (CQM) Hardware**	
QoS Engine (QoS) Hardware		
No. of Instances	1	
No. of Queues	512	
No. of Tx Port	256	
No. of QoS Scheduler Layers	7	
No. of Shared Rate Shapers, Each Assigned to 8 Nodes or Less	512	
No. of Schedulers	2048	
No. of Queue Depth for a Single Q	1M	
No. of Total Entries Supported	1M	
On-chip Segmented Buffer Memory Hardware		
No. of Instances	1	
On-chip Buffer Size	512 kB	
DDR Buffer Manager Hardware		
No. of Instances	1	
Buffer Start Address Alignment	128B	
No. of Supported Allocating Buffers per Second and Recycling Buffers per Second, Regardless of where the Pool Resides and of the DDR Memory Load	30M	15M
Pointer Towards DDR	36-bit	
No. of Pools	16	
Minimum Guarantee per Policy	A policy supports minimum guaranteed buffers from all of its related pools aggregated	
Maximum Allowed per Policy per Pool	A policy supports maximum allowed buffers per pool in the policy	
TEE		
Trusted Execution Processor Subsystem and Related Modules	**Trusted Execution Processor**	
Mailbox to TEE	MPS v1.0 (**Multi Processor System**)	
FIPS140-2 Level 2 Key Security Functions	Authentication, Asset Store, exclusive OTP access	
Signature Algorithm	• 1024 to 3072 bits • ECDSA: (FIPS186-4) Verification: P-192 to P-521 bits • ECDSA: (FIPS186-4) Generation P-224 to P-521 bits	

Table 4 Feature Overview - Interface Modules (cont'd)

Features	URX851/URX850	MxL25641
Encryption/Decryption Algorithm	• AES ECB, CBC, CTR (FIPS197, SP800-38A) CCM (SP800-38C) XTS (SP800-38E) CMAC (SP800-38B) 128, 192, 256 bits • TDES EBC, CBC (SP800-67, SP800-38A) 168 bits	
Message Digest/ MAC Generation/Verification	• SHA-1 (FIPS180-4), HMAC-SHA-1 [FIPS198-1): 112 to 512 bits • SHA-224 (FIPS180-4), HMAC-SHA-224 (FIPS198-1): 112 to 512 bits • SHA-256 (FIPS180-4), HMAC-SHA-256 (FIPS198-1): 128 to 512 bits • SHA-384 (FIPS180-4), HMAC-SHA-384 (FIPS198-1): 192 to 512 bits • SHA-512 (FIPS180-4), HMAC-SHA-512 (FIPS198-1): 256 to 512 bits	
Key Wrapping	• AES EBC with and without padding SP800-38F, RFC3394, RFC5649: 128, 192, 256 bits • With AES-CMAC, AES-CTR SP800-38B, SP800-38A: 512 bits (2x 256 bits)	
Key-based Key Derivation Function (KBKDF)	HMAC-SHA-256 (SP800-108): 128 to 512 bits	
Secure Boot	Authentication and decryption without external CPU/ROM code involvement	
OTP Version Key Security Functions	• One-time electrically field programmable non-volatile CMOS embedded memory • Voltage tamper detection • Signal tamper detection • Pseudo random data is inserted between read cycles to obfuscate hinder data probing or power analysis	
Crypto Engine Hardware (VPN Accelerator)[3]		
Encrypt/Decrypt Performance	5 Gbps/5 Gbps or asymmetric split Aggregated maximum 10 Gbps	2.5 Gbps/2.5 Gbps or asymmetric split Aggregated maximum 5 Gbps
IPsec ESP RFC 2403/4/5, RFC 2410, RFC 2451, RFC 1321, RFC 2104, RFC 4430, FIPS 46-3/81/180-1		
ESP Transform Using DES, Triple-DES, AES, or Null Encryption in CBC Mode with IPsec Padding and MD-5, SHA-1, or Null Authentication	AES encryption in ECB and CBC modes: 128, 192, 256 bits DES/TDES encryption in ECB, CBC: 168 bits	
ESP Outbound Header and Trailer Insertion Including Automatic Replay Counter Maintenance and HMAC ICV Calculation	RFC 2401, RFC 2403, RFC 2404, RFC 4430	
ESP Inbound Header and Trailer Validation Including SPI Comparison, Replay Counter (64-bit window) Processing, and HMAC ICV Verification	RFC 2401, RFC 2403, RFC 2404, RFC 4430	

Table 4 Feature Overview - Interface Modules (cont'd)

Features	URX851/URX850	MxL25641
IPsec ESP Transforms Using AES-GCM and AES-CCM	RFC 4106, RFC 4309	
IPsec ESP Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec	RFC4868	
Automatic IV Generation and Insertion	RFC 2405, RFC 2406, FIPS 46-3, FIPS 81, Draft FIPSAES, draft-ietf-IPSec-ciph-aes-cbc-01	
ESP Confidentiality RFC2451, RFC2405	DES(TDES)-CBC	
ESP Confidentiality RFC3602	AES-CBC with 128, 192, and 256-bit keys	
ESP Confidentiality RFC3686	AES-CTR with 128, 192, and 256-bit keys	
ESP Integrity RFC2403	HMAC-MD5	
ESP Integrity RFC2404	HMAC-SHA1	
ESP Integrity RFC4643	HMAC-SHA256/SHA-224 – SHA512/SHA384	
ESP Integrity RFC3566	XCBC-MAC-96	
ESP Combined Modes RFC4106, RFC4869, RFC6379	AES-GCM	
ESP Combined Modes RFC4543, RFC4869, RFC6379	AES-GMAC	
ESP Combined Modes RFC4309	AES-CCM with where A0 and B0 vectors calculated externally. Supports L value of 4 octets and key length of 128, 192, and 256-bit	
ESP ICV RFC4303	• Outbound: Appending of any length • Inbound: Checking any length	
SSL RFC6101		
Header Processing	• Outbound: Insertion type, version • Inbound: Removal of type and version	
Sequence Number	Generation. 64-bit overflow check	
Length field Processing	• Outbound: Fragment length is calculated by the host processor • Inbound: Payload length is calculated by the host processor for stream ciphers. For block ciphers, the payload length is calculated internally and is autonomously inserted in the data stream by the crypto engine. The last two cipher data blocks must be part of the provided processing token.	
Implicit IV	From the context, based on chained IVs	
Padding	• Inbound: Removal (and verification when the padding is constant) of SSL padding. The host determines and checks the pad length. • Outbound: Insertion of SSL padding from 0 to 255 (256 in total). The host calculates the length of padding.	
Cipher	• null-crypto • ARC4 • (T)DES-CBC • AES-CBC with 128, 192 and 256-bit keys	
Hash	• SSL-MAC-MD5 • SSL-MAC-SHA1	

Table 4 Feature Overview - Interface Modules (cont'd)

Features	URX851/URX850	MxL25641
ICV	• Outbound: Insertion • Inbound: Verification	
TLS RFC4346, RFC5246, RFC6460		
Header Processing	• Outbound: Insertion type, version • Inbound: Removal of type and version	
Sequence Number	Generation. 64-bit overflow check	
Explicit IV Processing for TLS 1.1 and 1.2	• Outbound: Inserting explicit IV from context or generated IV by the PRNG • Inbound: Retrieving explicit IV from input	
Explicit IV Processing for TLS 1.3	• Outbound: Inserting explicit IV from context or generated IV by the PRNG • Inbound: Retrieving explicit IV from input	
Cipher	• null-crypto • ARC4 • (T)DES-CBC • AES-CBC with 128, 192 and 256-bit keys • Stream cipher ChaCha20 with Poly1305 message authentication	
Hash	• SSL-MAC-MD5 • SSL-MAC-SHA1 • HMAC-SHA-256 (TLS 1.2, 1.3 only) • HMAC-SHA-384/512 (TLS 1.2, 1.3 only)	
Combined Modes	AES-GCM	
TLS ICV	• Outbound: Insertion • Inbound: Verification	
DTLS		
Header Processing	RFC4347, RFC6347 • Outbound: Insertion of type, version, epoch, generated sequence number • Inbound: Removal of type and version Removal and checking: epoch and sequence number	
Explicit Sequence Numbering	RFC4347, RFC6347 • Outbound: Generation. Overflow check of 48-bit • Inbound: IPsec type of replay check of 48-sequence number with 64 or 128 bit mask.	
Epoch	RFC4347, RFC6347 • Outbound: Insertion • Inbound: Check	
Fragment Compression/Decompression	RFC4347, RFC6347	
Length Field Processing	RFC4347, RFC6347 • Outbound: Fragment length is calculated by the host processor. • Inbound: Payload length is calculated by the host processor for stream ciphers. For block ciphers, the payload length is calculated internally and is autonomously inserted in the data stream. The last two cipher data blocks must be part of the processing token provided to the crypto engine.	
IV Processing	RFC4347, RFC6347 • Outbound: Insertion of explicit IV from the context • Inbound: Taking explicit IV from the input	

Table 4 Feature Overview - Interface Modules (cont'd)

Features	URX851/URX850	MxL25641
Padding	RFC4347, RFC6347 Same as in TLS 1.1	
Cipher	RFC4347, RFC6347 • null-crypto • (T)DES-CBC • AES-CBC with 128, 192 and 256-bit keys	
Hash	RFC4347, RFC6347 • SSL-MAC-MD5 • SSL-MAC-SHA1 • HMAC-SHA-256 (TLS 1.2 only) • HMAC-SHA-384/512 (TLS 1.2 only)	
Combined Modes	RFC4347, RFC6347 • AES-GCM	
ICV	RFC4347, RFC6347 • Outbound: Insertion • Inbound: Verification	
SRTP/SRTCP RFC3711		
UDP Header	• Outbound: Bypass • Inbound: Bypass	
MKI Field (Optional)	• Outbound: Insertion from context • Inbound: Removal, verification	
SRTP ROC	Used from token (calculated by the host)	
SRTCP E+Index	• Outbound: Insertion (from token) • Inbound: Removal	
IV Processing (Defined Externally)	From context (IV is calculated by the host)	
Cipher	• null-crypto • AES-ICM (overflow of the 16-bit counter to be checked by the host)	
Hash	HMAC SHA1	
TAG (Variable Length)	• Outbound: Insertion • Inbound: Verification	
Wireless Mode		
SNOW 3G	128-EEA1 and 128-EIA1 (basic mode)	-
Kasumi	Basic, f8, and f9 (basic mode)	-
ZUC	128-EEA3 and 128-EIA3 (basic mode)	-
MACsec IEEE 802.1AE-2006, IEEE 802.1AEbn-2011 (Look-aside Engine)		
Header Processing	• Outbound: Insertion of SPI token, PN and SCI from context • Inbound: Removal	
IV Processing	• Outbound: From context • Inbound: – From input header (SCI), or – From input header and context (without SCI)	

Table 4 Feature Overview - Interface Modules (cont'd)

Features	URX851/URX850	MxL25641
Packet Number	• Outbound: Generation and overflow check • Inbound: Verification and in-window check (32-bit integer)	
ICV (16-byte)	• Outbound: Insertion • Inbound: Verification	
Confidentiality Offset	• Outbound: Supported • Inbound: Supported	
ICU (Interrupt Control Unit)		
No. of Instances	1	
No. of Interrupts	224 - (0-31 reserved for the internal Intel Atom subsystem)	184 - (0-31 reserved for the internal Intel Atom subsystem)
MSI Direct to Core	MSI mapped to core via IO coherency bridge. Up to 32 MSI supported.	
No. of External Interrupt	All General Purpose Input/Output (GPIO) mapped to interrupt inputs	
Distributes/Partitions the Interrupt Sources	Among the available core (via internal MSI)	
Steers any Interrupt Source	To any core (VMs require hypervisor/VMM to manage interrupt low level)	
CPU/VM Cross Interrupt	Allows VM to VM interrupt	
I^2C		
No. of Instances	4	
Speed Modes	• Standard mode (<100 Kb/s) • Fast mode (<400 Kb/s) • Fast mode plus (<1000 Kb/s) • High speed mode (<3.4 Mb/s)	
Watchdog Timer (WDT)		
No. of Instances	TEE/TEP: 1 WDT 1 WDT set for the Intel Atom subsystem via GPTC timers (2 WDT - pre-reset warning and reset GPTC timers per core)	
No. of GPTC Timers as WDT	8	4
UART (ASC used by Intel Atom)		
No. of Instances	4	
Baud Rates Supported	426 Baud - 12.6 MBaud[4)]	
Voice DSP		
Operating Frequency with Active Voice	1. 393 MHz 2. 524 MHz (default) 3. 629 MHz	
Voice Mailbox towards Linux OS	DSP to Intel Atom Mailbox Method	
PCM + DMA Flow Control Controller (DMA FCC)		
No. of PCM Instances	3	
Supported PCM Clock Rates (Clock Master) (MHz)	• 0.512 • 1.536 • 2.048 • 4.096 • 8.192	

Table 4 Feature Overview - Interface Modules (cont'd)

Features	URX851/URX850	MxL25641
8 kHz PCM Frame Sync Clock Supported with Programmable High Phase	3 DCL clocks or 50% FSC period	
No. of DMA FCC Instances	3	
Allows Setting 2x Buffers (PING, PONG) and Uses these 2 in Alternating Way to Collect/Send PCM Samples to PCM Module	Supported	
V-CODEC Module		
No. of Instances	2	2
No. of Voice Channels	4	4
I2S		
FIFO Size	8 x 32-bit	
No. of Instances	2	
SPI (SSC Top Level Intel Atom)		
No. of Instances	4	
Support to Connect External Devices such as DECT CatIQ, Serial Flash or Serial EEPROM	• SPI0: maximum 3 • SPI1: maximum 6 • SPI2: maximum 3 • SPI3: maximum 2	
FIFO Size	64 x 32-bit	
Interface Clock Master Mode	50 MHz	
Interface Clock Slave Mode	25 MHz	
Quad SPI		
No. of Instances	1	
Support to Connect External Devices Serial Flash or Serial EEPROM	2	
Interface Clock Master Mode	100 MHz	
General Purpose Timer/Counter (GPTC)		
No. of Instances	4	
No. of 16/32-bit Timers *Note: It is possible to combine 2x 16-bit timers into a 1x 32-bit timer*	24/12 (4 instances of GPTC)	
Used as Internal Timer with Interrupt Generation	Supported	
Detection of External Logic Level and Generation of Interrupt	Supported	
Generation of External Clocks via Counter Mode	Supported	

Table 4 Feature Overview - Interface Modules (cont'd)

Features	URX851/URX850	MxL25641
Generation of Interrupt Signal Based on Timer Used as NMI for Intel Atom CPU Watchdog Functionality	Supported	
External Interrupt		
External Interrupt (EXIN) Interface Inputs	Every GPIO pin **Connecting Interrupt to IOAPIC**	
Output Clocks		
Clock Output 1	Supports 40 MHz (XTAL clock).	
Clock Output 2	Supports 25 MHz.	
Timing Synchronization		
PPS and IEEE1588	Supported	
GPIO		
GPIO Multiplexing Features	**Project specific multiplexing**	
Serial Shift Output Controller - Part of GPIO Module	• Individual duty cycle control • Separated blinking rate control • LED dimming with off-chip brightness detector	
Fan Controller	**PWM FAN Controller**	
MDIO Controller	**2**	
Management/Control		
UART Controller for RS-232 ASC (UART) Controller	Supported	
EJTAG/JTAG		
Core Debug Interface	JTAG	
Boundary Scan JTAG Interface (IEEE 1149.1 Compliant)	Supported	
Secure EJTAG	• Crypto authentication • Permanent disable	
Analog Modules		
Power-on Reset (POR) Detection Including Under Voltage Detection (UVD)	Supported	
PVT Monitor	Supported	
XO	Supported	

1) Voltage must be 0.9 V.
2) Up to additional 4 PCIe interfaces can be supported via the multiplexed Combo HSIO Subsystem.
3) From a security standpoint, higher key/cipher/digest lengths like ECDSA-256/384, AES-256, SHA-256, TLS1.2 and/or their combinations are required, and software releases may not support deprecated security modes.
4) Based on 200 MHz UART controller clock. This is the hardware interface's baud-rate, which may not be attainable at the system level because the CPU is likely unable to serve associated interrupts at this rate.

1.3 System Considerations

URX851, URX850, and MxL25641 are designed to be flexibly used and to provide a variety of different interface combinations. To keep the package and pin count small, various functionalities are multiplexed for certain pins.

Low speed interfaces are multiplexed on GPIO pins. See **Table 9** for more details.

Depending on SKU capabilities, it is possible to multiplex some high speed interfaces (SerDes). See **Table 89** and **Table 92** for more details.

Note: Some high speed SerDes support dual configuration. The multiplexing of these SerDes to different interface technologies must be done in pairs. For example, HSIO module 1 has 2 lanes which must be configured for the same protocol. In other words, both lanes are PCIe, or both lanes are SATA, or both lanes are XFI Ethernet.

For XFI Ethernet, these combinations are concurrently supported for both lanes of HSIO 1 and 2:

Table 5 XFI Ethernet Modes (Applicable Only for URX851/URX850/MxL25641)

Lane 1 Mode ==>	**2.5G-XGMII**	**1000BASE-KX/ SGMII**	**2.5G-SGMII (8B/10B)**	**10G-KR/10G/5G/ 2.5G-USXGMII**
Lane0 Mode				
2.5G-XGMII	Yes	Yes	Yes	No
1000BASE-KX/SGMII	Yes	Yes	Yes	Yes
2.5G-SGMII(8b/10b)	Yes	Yes	Yes	No
10G-KR/10G/5G/2.5G-USXGMII	No	Yes	No	Yes

2 Pin Description of URX851/URX850

This chapter describes the pin diagram for URX851/URX850.

2.1 List of Interfaces

Table 6 summarizes the supported interfaces.

Table 6 External Signals

Interface	Link
PCIe 4.0/XFI/SATA Lane 0/1/2/3	Table 10 Table 11 Table 12 Table 13
PCIe 3.0 Lane 4/5/6/7	Table 14 Table 15 Table 16 Table 17
USB 3.2 Port 0/Port1	Table 21
GPIO Port includes: • LED Controller • External Interrupts • System Clocks • 16/8-bit NAND • PCM • SPI • I2S • I^2C • QSPI	Table 22
JTAG	Table 24
32-bit DDR3L/4/LPDDR4 SDRAM	Table 25
Clock	Table 26
Reset	Table 27
Power	Table 28
Fuse	Table 30
Ethernet Subsystem Signals	Table 31
Ethernet Media Interface Signals	Table 34

2.2 Ball Diagram for URX851/URX850

The color code means:

- Green = DDR Interface
- Orange = Power
- Red = WAN Signals
- Light Red = PCIe
- Blue = XFI/SFI/SGMII Signals
- Pink = Clock, Reset, Dying Gasp
- Yellow = GPIO, eMMC, EJTAG Interface
- Grey = Ground, Fuse, Reserved
- White = NC

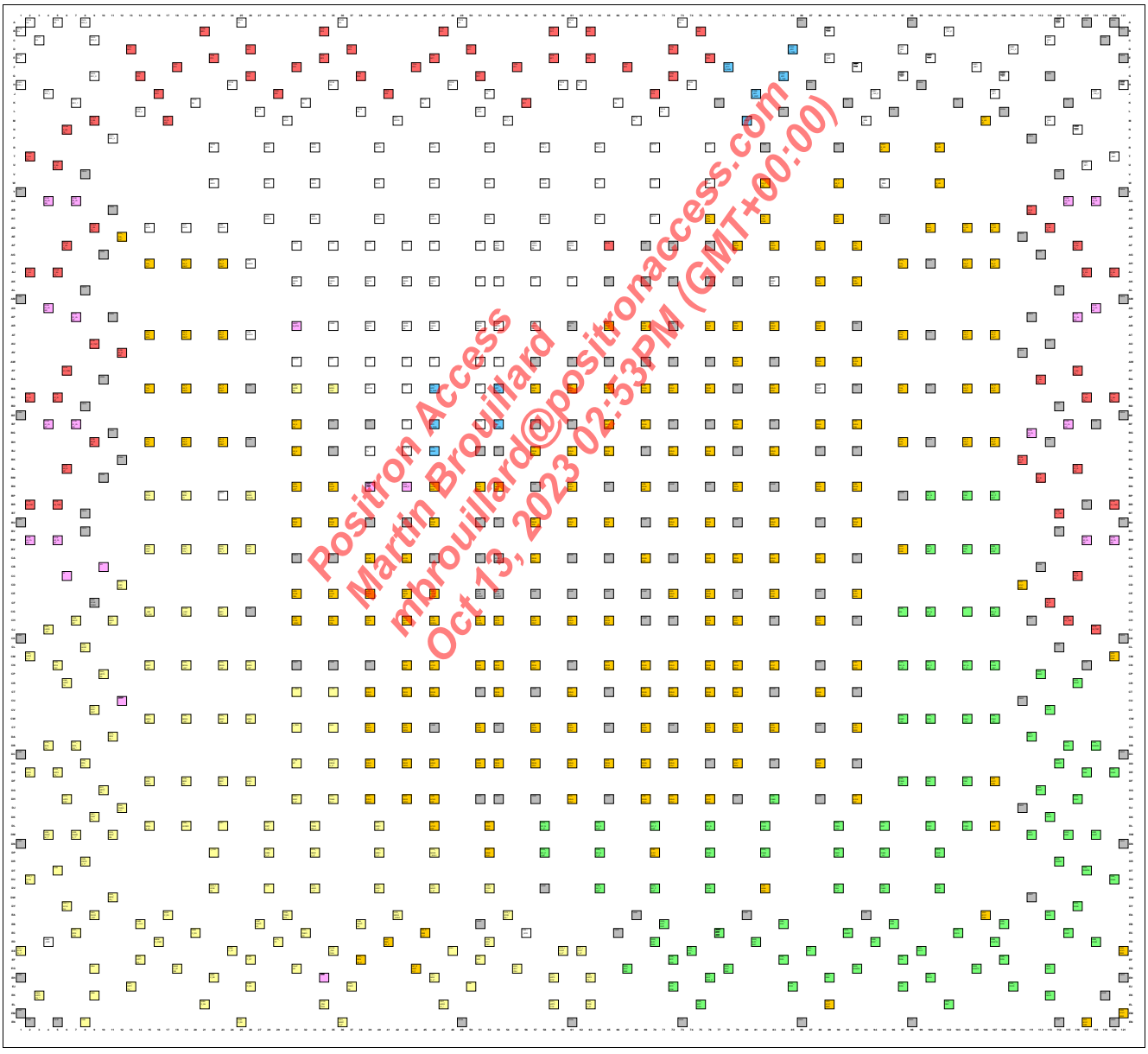


Figure 5 Ball Diagram for PG-FCBGA-840 (Top View)

2.3 Ball (Pin) Function Description

Use these abbreviations for the I/O table.

Table 7 Abbreviations for Pin Type

Abbreviations	Description
I	Input only, digital levels
O	Output only, digital levels
I/O	Bidirectional input/output signal, digital levels
AI	Input only, analog levels
AO	Output only, analog levels
AI/O	Bidirectional, analog levels
PWR	Power
GND	Ground
MCL	Must be connected to Low (JEDEC Standard).
MCH	Must be connected to High (JEDEC Standard).
NU	Not Usable (JEDEC Standard)
NC	Not Connected (JEDEC Standard). Do not connect in board design.
Prg	Programmable (either input or output)

Table 8 Abbreviations for Buffer Type

Abbreviations	Description
A	Analog
Z	High impedance
Prg	Programmable (OD/PP, PU/PD are programmable)
PU1	Pull up (internal, weak)
PD1	Pull down (internal, weak)
PD2	Pull down (internal, weaker)
TS	Tristate capability: The corresponding pin has 3 operational states: Low, high and high-impedance.
OD	Open Drain. The corresponding pin has 2 operational states, active low and tristate, and allows multiple devices to share as a wire-OR. An external pull-up is required to sustain the inactive state until another agent drives it, and must be provided by the central resource.
OC	Open Collector
PP	Push-Pull. The corresponding pin has 2 operational states: Active-low and active-high (identical to output with no type attribute).
OD/PP	Open-Drain or Push-Pull. The corresponding pin can be configured either as an output with the OD attribute or as an output with the PP attribute.
ST	Schmitt-Trigger characteristics
TTL	TTL characteristics

Table 9 GPIO Interface Multiplexing

The individual pin function after reset is flagged by a note ('Note: Default after Reset.'), otherwise it is the first column field, in other words GPIO function.

PORTMUX bit [00] 0	PORTMUX bit [01] 1	PORTMUX bit [10] 2	PORTMUX bit [11] 3
IO00_1588GPC1_SPI1CS2	GPC1_1588	–	SPI1_CS2_N
IO01_SPI2CS2	USB_C_PWR_EN1	–	SPI2_CS2_N
IO02_SPI1CS4	USB_C_PWR_EN2	–	SPI1_CS4_N
IO03_CKO2_I2S1CLK	CLKOUT2	–	I2S_CLK1
GPIO4	LED_ST	LED_DD0	IO04_LEDST_LEDDD0_SPISLVCK *Note: Default after Reset.*
GPIO5	LED_D	LED_DD1	IO05_LEDD_LEDDD1_SPISLVI *Note: Default after Reset.*
GPIO6	LED_SH	LED_DD2	IO06_LEDSH_LEDDD2_SPISLVO *Note: Default after Reset.*
IO07_LEDSH1_CKOUT1_FANIN	LED_SH1	CLKOUT1	FAN_CTRL_I
IO08_VSPI1CS1_ZSIFSC_CKO0	C55_SPI1_CS1	CLKOUT0	-
GPIO9	SPI2_CS1_N	PON_Tx_SD	IO09_SPI2CS1 *Note: Default after Reset.*
IO10_URT2TX_SPI1TX_SSI1TX	SPI1_TX	SSI1_TX	UART2_TX
IO11_URT2RX_SPI1RX_SSI1RX	SPI1_RX	SSI1_RX	UART2_RX
GPIO12	–	IO12_I2C0SDA-PMIC *Note: Default after Reset.*	SPI0_CS6_N
GPIO13	IO13_NDALE_SPI1CS3 *Note: Default after Reset.*	--	SPI1_CS3_N
IO14_URT2RTS_SPI1CS0_VRST1	SPI1_CS0_N	SLIC_RST1	UART2_RTS
IO15_SPI0CS1_SPI1CS0	SPI0_CS1_N	–	C55_SPI1_CS0
IO16_SPI0DI_SPI1RX	SPI0_DIN	–	C55_SPI1_RX
IO17_SPI0DO_SPI1TX	SPI0_DOUT	–	C55_SPI1_TX
IO18_SPI0CK_SPI1CK	SPI0_CLK	–	C55_SPI1_CLK
IO19_URT2CTS_SPI1CK_SSI1CK	SPI1_CLK	SSI1_CLK	UART2_CTS
GPIO20	–	IO20_I2C0SCL-PMIC *Note: Default after Reset.*	SPI1_CS5_N

Table 9 GPIO Interface Multiplexing (cont'd)

The individual pin function after reset is flagged by a note ('Note: Default after Reset.'), otherwise it is the first column field, in other words GPIO function.

PORTMUX bit [00] 0	PORTMUX bit [01] 1	PORTMUX bit [10] 2	PORTMUX bit [11] 3
IO21_I2C1SDA	–	I2C_SDA1	–
IO22_I2C1SCL	–	I2C_SCL1	–
GPIO23	IO23_NDCS1_LEDST1 *Note: Default after Reset.*	LED_ST1	–
GPIO24	IO24_NDCLE_SPI0CS4 *Note: Default after Reset.*	–	SPI0_CS4_N
GPIO25	IO25_NDBC0_1588GPC2 *Note: Default after Reset.*	–	GPC2_1588
GPIO26	IO26_NDBC1 *Note: Default after Reset.*	–	–
IO27_SPI1CS1_VRST0	SLIC_RST0	SPI1_CS1_N	C55_RESET0
IO28_ZSIFSC_TDM1FSC	-	TDM1_FSC	–
IO29_ZSIM2S_TDM1DO	-	TDM1_DO	–
IO30_ZSIS2M_TDM1DI	-	TDM1_DI	–
IO31_ZSIDCL_TDM1DCL	-	TDM1_DCL	–
IO32_LEDD1_TDM0FSC_I2S1WA	LED_D1	I2S_WA1	TDM0_FSC
IO33_TDM0DO_I2S1TX		TDM0_DO	I2S_TX1
IO34_VSPI1TX_ZSIM2S_SSI0TX	SSI0_TX	C55_SPI1_TX	-
IO35_VSPI1RX_ZSIS2M_SSI0RX	SSI0_RX	C55_SPI1_RX	-
IO36_VSPI1CK_ZSIDCL_SSI0CK	SSI0_CLK	C55_SPI1_CLK	-
IO37_TDM0DI_I2S1RX	–	TDM0_DI	I2S_RX1
IO38_TDM0DCL_I2S1CK	–	TDM0_DCL	I2S_CLK1
IO39_SPI3CS1	PON_LOS	SPI3_CS1_N	–
IO40		LED_DD3	–
IO41_MDIO0CKWAN_FANIN	MDIO0_WANC	FAN_CTRL_IN	LED_DD4
IO42_MDIO0DWAN_FANOUT	MDIO0_WAN	LED_DD5	FAN_CTRL_OUT
IO43	–	SATA_MP_Switch	–
IO44_I2S0CK_SPI2CK	I2S_CLK0	SPI2_CLK	–
IO45_I2S0WA_SPI2TX	I2S_WA0	SPI2_TX	–
IO46_I2S0RX_SPI2RX	I2S_RX0	SPI2_RX	–

Table 9 GPIO Interface Multiplexing (cont'd)

The individual pin function after reset is flagged by a note ('Note: Default after Reset.'), otherwise it is the first column field, in other words GPIO function.

PORTMUX bit [00] 0	PORTMUX bit [01] 1	PORTMUX bit [10] 2	PORTMUX bit [11] 3
IO47_I2S0TX_SPI2CS0	I2S_TX0	SPI2_CS0_N	–
GPIO48	IO48_NDRDBYN *Note: Default after Reset.*	–	–
GPIO49	IO49_NDRDN *Note: Default after Reset.*	–	–
GPIO50	IO50_NDD1_QSPID1 *Note: Default after Reset.*	–	QSPI_D1
GPIO51	IO51_NDD0_QSPID0 *Note: Default after Reset.*	–	QSPI_D0
GPIO52	IO52_NDD2_QSPID2 *Note: Default after Reset.*	–	QSPI_D2
GPIO53	IO53_NDD7_QSPID3 *Note: Default after Reset.*	–	QSPI_D3
GPIO54	IO54_NDD6_QSPICK *Note: Default after Reset.*	–	QSPI_CLK
GPIO55	IO55_NDD5_QSPIRST0 *Note: Default after Reset.*	–	QSPI_RST0
GPIO56	IO56_NDD4_QSPICS0 *Note: Default after Reset.*	–	QSPI_CS0
GPIO57	IO57_NDD3_QSPICS1 *Note: Default after Reset.*	–	QSPI_CS1
GPIO58	IO58_NDCS0_QSPIRST1 *Note: Default after Reset.*	–	QSPI_RST1
GPIO59	IO59_NDWRN *Note: Default after Reset.*	–	–
GPIO60	IO60_NDWPN_VINT1_I2STX1 *Note: Default after Reset.*	SLIC2C55_INT1	I2S_TX1
GPIO61	IO61_NDSE_VINT0_I2SWA1 *Note: Default after Reset.*	SLIC2C55_INT0	I2S_WA1
IO62_I2C3SCL_I2SRX1	I2C_3_SCL	–	I2S_RX1
IO63_I2C3SDA	I2C_3_SDA	–	–
GPIO64	IO64_URT0RX *Note: Default after Reset.*	–	–

Table 9 GPIO Interface Multiplexing (cont'd)

The individual pin function after reset is flagged by a note ('Note: Default after Reset.'), otherwise it is the first column field, in other words GPIO function.

PORTMUX bit [00] 0	PORTMUX bit [01] 1	PORTMUX bit [10] 2	PORTMUX bit [11] 3
GPIO65	IO65_URT0TX *Note: Default after Reset.*	–	–
IO66_URT1RX_TDM2FSC	UART1_RX	C55_TDM2_FSC	TDM2_FSC
IO67_URT1TX_TDM2DO	UART1_TX	C55_TDM2_DO	TDM2_DO
IO68_URT2RX_TDM2DI	UART2_RX	C55_TDM2_DI	TDM2_DI
IO69_URT2TX_TDM2DCL	UART2_TX	C55_TDM2_DCL	TDM2_DCL
GPIO70	I2C_SDA2	IO70_I2C2DA_URT3RX_SPI3CS0 *Note: Default after Reset.*	SPI3_CS0_N
GPIO71	I2C_SCL2	IO71_I2C2CK_URT3TX_SPI3TX *Note: Default after Reset.*	SPI3_TX
GPIO72	TX_Fault	IO72_URT3RTS_SPI3RX *Note: Default after Reset.*	SPI3_RX
GPIO73	TX_DISABLE	IO73_URT3CTS_SPI3CK *Note: Default after Reset.*	SPI3_CLK
GPIO74	IO74_NDD13_SDCK *Note: Default after Reset.*	SATA0_PHY_DEVSLP	SDIO_CLK
GPIO75	IO75_NDD12_SDCMD *Note: Default after Reset.*	SATA1_PHY_DEVSLP	SDIO_CMD
GPIO76	IO76_NDD10_SDD2 *Note: Default after Reset.*	SATA2_MP_SWH	SDIO_D2
GPIO77	IO77_NDD11_SDD3 *Note: Default after Reset.*	SATA2_CP_DET	SDIO_D3
GPIO78	IO78_NDD9_SDD1 *Note: Default after Reset.*	SATA3_MP_SWH	SDIO_D1
GPIO79	IO79_NDD8_SDD0 *Note: Default after Reset.*	SATA3_CP_DET	SDIO_D0
GPIO80	IO80_NDD14_SDCD *Note: Default after Reset.*	SATA1_MP_SWH	SDIO_CD
GPIO81	IO81_NDD15_SDWP *Note: Default after Reset.*	SATA1_CP_DET	SDIO_WP
IO82_LEDBD	–	SATA0_CP_DET	LED_BD
IO83	–	PCI10_CLK_REQ[1)]	–
IO84	–	PCI11_CLK_REQ[1)]	–

Table 9 GPIO Interface Multiplexing (cont'd)

The individual pin function after reset is flagged by a note ('Note: Default after Reset.'), otherwise it is the first column field, in other words GPIO function.

PORTMUX bit [00] 0	PORTMUX bit [01] 1	PORTMUX bit [10] 2	PORTMUX bit [11] 3
IO85	–	PCI20_CLK_REQ[1)]	–
IO86	–	PCI21_CLK_REQ[1)]	–
IO87	–	PCI30_CLK_REQ[1)]	–
IO88	–	PCI31_CLK_REQ[1)]	–
IO89	–	PCI40_CLK_REQ[1)]	–
IO90_PMICIO5	PMIC_GPIO5	PCI41_CLK_REQ[1)]	–
IO91_PPSI-O_LEDDD6	PPS1_IN_OUT	LED_DD6	–
IO92_PPSO-I_LEDDD7	PPS2_IN_OUT	LED_DD7	–
IO93_CK32K_NTR_LEDDD9	CLK32K	NTR	LED_DD9
IO94_MDIO1CKLAN	MDIO1_CLK	–	–
IO95_MDIO1DLAN	MDIO1_D	–	–
GPIO96	IO96_PMICIO0 *Note: Default after Reset.*	–	–
GPIO97	IO97_PMICIO1 *Note: Default after Reset.*	–	–
GPIO98	IO98_PMICIO2 *Note: Default after Reset.*	–	–
IO99_LEDDD8		LED_DD8	Sata2_PHY_devslp
IO100_VSCC1RST1	C55_SCC1_RESET1	–	–
IO101_VSPI0_CS1	C55_SPI0_CS1	–	Sata3_PHY_devslp
GPIO102	IO102_PMICIO3 *Note: Default after Reset.*	–	–
GPIO103	IO103_PMICIO4 *Note: Default after Reset.*	–	–

1) This pin can be used to give HW indication of PCI clock request to this particular lane when programmed in this alternative function mode; otherwise, this pin is a general GPIO.

2.3.1 PCIe 4.0/SATA 3.2/XFI Interface #10 of HSIO1

Table 10 describes the PCI Express interface. These pins are multiplexable with 10G XFI or SATA interface pins. For details on Ethernet usage, see PCIe 4.0/SATA 3.2/XFI Combo Subsystem.

Table 10 PCIe 4.0/SATA 3.2/XFI Interface #10 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
AJ120	**PCIe10_SATA_XPCS_TXP**	AO	–	**Transmit Data Pair (Differential)**[1]
AJ117	**PCIe10_SATA_XPCS_TXN**	AO	–	
AF116	**PCIe10_SATA_XPCS_RXP**	AI	–	**Receive Data Pair (Differential)**
AD113	**PCIe10_SATA_XPCS_RXN**	AI	–	
AB111	**PCIE10_RREF**	AI/AO	–	**Reference Resistor for HSIO module 1** *Note: 1% accuracy required.*

1) This SATA interface is for SATA0 of module 1.

2.3.2 PCIe 4.0/SATA 3.2/XFI Interface #11 of HSIO1

Table 11 describes the PCI Express interface. These pins are multiplexable with 10G XFI or SATA interface pins. For details on Ethernet usage, see PCIe 4.0/SATA 3.2/XFI Combo Subsystem.

Table 11 PCIe 4.0/SATA 3.2/XFI Interface #11 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
BA112	**PCIe11_SATA_XPCS_TXP**	AO	–	**Transmit Data Pair (Differential)**[1]
AY116	**PCIe11_SATA_XPCS_TXN**	AO	–	
BC120	**PCIe11_SATA_XPCS_RXP**	AI	–	**Receive Data Pair (Differential)**
BC117	**PCIe11_SATA_XPCS_RXN**	AI	–	

1) This SATA interface is for SATA1 of module 1.

2.3.3 PCIe 4.0/SATA 3.2/XFI Interface #20 of HSIO2

Table 12 describes the PCI Express interface. These pins are multiplexable with 10G XFI or SATA interface pins. For details on Ethernet usage, see PCIe 4.0/SATA 3.2/XFI Combo Subsystem.

Table 12 PCIe 4.0/SATA 3.2/XFI Interface #20 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
N6	**PCIe20_SATA_XPCS_TXP**	AO	–	**Transmit Data Pair (Differential)**[1]
M9	**PCIe20_SATA_XPCS_TXN**	AO	–	
T2	**PCIe20_SATA_XPCS_RXP**	AI	–	**Receive Data Pair (Differential)**
U5	**PCIe20_SATA_XPCS_RXN**	AI	–	
M17	**PCIE20_RREF**	AI/AO	–	**Reference Resistor for HSIO module 2** *Note: 1% accuracy required.*

1) This SATA interface is for SATA0 of module 2.

2.3.4 PCIe 4.0/SATA 3.2/XFI Interface #21 of HSIO2

Table 13 describes the PCI Express interface. These pins are multiplexable with 10G XFI or SATA interface pins. For details on Ethernet usage, see PCIe 4.0/SATA 3.2/XFI Combo Subsystem.

Table 13 PCIe 4.0/SATA 3.2/XFI Interface #21 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
AJ5	**PCIe21_SATA_XPCS_TXP**	AO	–	**Transmit Data Pair (Differential)**[1]
AJ2	**PCIe21_SATA_XPCS_TXN**	AO	–	
AD9	**PCIe21_SATA_XPCS_RXP**	AI	–	**Receive Data Pair (Differential)**
AF6	**PCIe21_SATA_XPCS_RXN**	AI	–	

1) This SATA interface is for SATA1 of module 2.

2.3.5 PCIe 3.0 Interface #30 of HSIO3

Table 14 describes the PCI Express interface.

Table 14 PCIe 3.0 Interface #30 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
CJ118	**PCIE30_TXP**	AO	–	**Transmit Data Pair (Differential)**
CH115	**PCIE30_TXN**	AO	–	
CF113	**PCIE30_RXP**	AI	–	**Receive Data Pair (Differential)**
CC116	**PCIE30_RXN**	AI	–	
BK110	**PCIE30_RREF**	AI/AO	–	**Reference Resistor for HSIO module 3** *Note: 1% accuracy required.*

2.3.6 PCIe 3.0 Interface #31 of HSIO3

Table 15 describes the PCI Express interface.

Table 15 PCIe 3.0 Interface #31 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
BT114	**PCIE31_TXP**	AO	–	**Transmit Data Pair (Differential)**
BR120	**PCIE31_TXN**	AO	–	
BM112	**PCIE31_RXP**	AI	–	**Receive Data Pair (Differential)**
BL116	**PCIE31_RXN**	AI	–	

2.3.7 PCIe 3.0 Interface #40 of HSIO4

Table 16 describes the PCI Express interface.

Table 16 PCIe 3.0 Interface #40 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
BR2	**PCIE40_TXP**	AO	–	**Transmit Data Pair (Differential)**
BR5	**PCIE40_TXN**	AO	–	
BL6	**PCIE40_RXP**	AI	–	**Receive Data Pair (Differential)**
BH9	**PCIE40_RXN**	AI	–	
AV12	**PCIE40_RREF**	AI/AO	–	**Reference Resistor for HSIO module 4** *Note: 1% accuracy required.*

2.3.8 PCIe 3.0 Interface #41 of HSIO4

Table 17 describes the PCI Express interface.

Table 17 PCIe 3.0 Interface #41 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
AU9	**PCIE41_TXP**	AO	–	**Transmit Data Pair (Differential)**
AY6	**PCIE41_TXN**	AO	–	
BC5	**PCIE41_RXP**	AI	–	**Receive Data Pair (Differential)**
BC2	**PCIE41_RXN**	AI	–	

2.3.9 PON_XFI Interface #4

Table 18 describes the PON_XFI interface.

For URX851, this interface is usable as PON or XFI (Ethernet) interface.

For URX850, this interface is XFI (Ethernet) interface only.

Table 18 PON_XFI#4 Interface Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
G84	**PON0_XPCS_TXP**	AO	–	**Transmit Data Pair (Differential)**
D85	**PON0_XPCS_TXN**	AO	–	
J81	**PON0_XPCS_RXP**	AI	–	**Receive Data Pair (Differential)**
F78	**PON0_XPCS_RXN**	AI	–	
M80	**PON0_RREF**	AI/AO	–	**Reference Resistor** *Note: 1% accuracy required.*

2.3.10 XFI Interface to the Ethernet PHYs

Table 19 describes the XFI interface.

Table 19 XFI Interface Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
BB53	**XFI5_XPCS_TXP**	AO	–	**Transmit Data Pair (Differential)**
BF53	**XFI5_XPCS_TXN**	AO	–	
BB46	**XFI5_XPCS_RXP**	AI	–	**Receive Data Pair (Differential)**
BF46	**XFI5_XPCS_RXN**	AI	–	
BJ46	**XFI5_RREF**	AI/AO	–	**Reference Resistor** *Note: 1% accuracy required.*

2.3.11 USB Port 0

Table 20 describes the USB0 interface for USB3.2 Gen2 x1, Type-C.

Table 20 USB0 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
T120	**USB0_DM1**	AI/AO	–	**D- for USB 2.0**
U117	**USB0_DP1**	AI/AO	–	**D+ for USB 2.0**
J118	**USB0_TX0_P**	AO	–	**Transmit Data Pair (Differential) for USB 3.2 Gen 2 x1**
H121	**USB0_TX0_N**	AO	–	
D109	**USB0_TX1_P**	AO	–	**Transmit Data Pair (Differential) for USB 3.2 Gen 2 x1**
G108	**USB0_TX1_N**	AO	–	
C113	**USB0_RX1_P**	AI	–	**Receive Data Pair (Differential) for USB 3.2 Gen 2 x1**
A114	**USB0_RX1_N**	AI	–	
M113	**USB0_RX0_P**	AI	–	**Receive Data Pair (Differential) for USB 3.2 Gen 2 x1**
N116	**USB0_RX0_N**	AI	–	
W95	**USB0_RREF0**	AI/AO	–	**Reference Resistor** *Note: 1% accuracy required. Applicable to USB 3.2 PHY Interface.*
BB88	**URESREFU20**	AI/AO	–	**Reference Resistor** *Note: 1% accuracy required. Applicable to USB 2.0 PHY Interface.*
ED4	**USB0_OCD**	AI	–	**USB Overcurrent Sensing Circuit 0 / Dying Gasp** The overcurrent detection measures the drop on the supply voltage. When the connected USB device draws too much power, the sensing circuit generates an interrupt and a signal. The CPU is able to switch off the power supply to the USB device in such an interrupt event. The signal goes to dedicated GPIO which is used to switch off the USB supply. Another alternative use for the OCD module is to create a dying gasp circuit as described in **Overcurrent Detection Comparator**.

2.3.12 USB Port 1

Table 21 describes the USB1 interface for USB3.2 Gen2 x1, Type-C.

Table 21 USB1 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
F105	**USB1_DM1**	AI/AO	–	**D- for USB 2.0**
J107	**USB1_DP1**	AI/AO	–	**D+ for USB 2.0**
J94	**USB1_TX1_P**	AO	–	**Transmit Data Pair (Differential) for USB 3.2 Gen 2 x1**
F92	**USB1_TX1_N**	AO	–	
E89	**USB1_RX1_P**	AI	–	**Receive Data Pair (Differential) for USB 3.2 Gen 2 x1**
B89	**USB1_RX1_N**	AI	–	
B102	**USB1_TX0_P**	AO	–	**Transmit Data Pair (Differential) for USB 3.2 Gen 2 x1**
E100	**USB1_TX0_N**	AO	–	
D97	**USB1_RX0_P**	AI	–	**Receive Data Pair (Differential) for USB 3.2 Gen 2 x1**
G97	**USB1_RX0_N**	AI	–	
M93	**USB1_RREF0**	AI/AO	–	**Reference Resistor** *Note: 1% accuracy required. Applicable to USB 3.2 PHY Interface.*
AK83	**URESREFU21**	AI/AO	–	**Reference Resistor** *Note: 1% accuracy required. Applicable to USB 2.0 PHY Interface.*
EC56	**USB1_OCD**	AI	–	**USB Overcurrent Sensing Circuit / Dying Gasp** The overcurrent detection measures the drop on the supply voltage. When the connected USB device draws too much power, the sensing circuit generates an interrupt and a signal. The CPU is able to switch off the power supply to the USB device in such an interrupt event. The signal goes to dedicated GPIO which is used to switch off USB supply. Another alternative use for the OCD module is to create a dying gasp circuit as described in **Overcurrent Detection Comparator**.

2.3.13 GPIO/I²C/I2S/SPI/PCM

Table 22 describes the GPIO interface.

Table 22 GPIO/Flash/I²C/I2S/SPI/PCM Signals

Ball No.	Name	Pin Type	Buffer Type	Function
DJ12	**IO00_1588GPC1_SPI1CS2**	Prg	Prg	**General Purpose I/O Function[P0.0]** This function is configurable via GPIO register.
	GPC1_1588	I/O		**IEEE1588 Clock Input/Output**
	SPI1_CS2_N	O		**SPI1 Chip Select 2**
EC7	**IO01_SPI2CS2**	Prg	Prg	**General Purpose I/O Function[P0.1]** This function is configurable via GPIO register.
	USB_C_PWR_EN1	O		**USB0 Port Power Enable**
	SPI2_CS2_N	O		**SPI2 Chip Select 2**
EG55	**IO02_SPI1CS4**	Prg	Prg	**General Purpose I/O Pin[P0.2]** This function is configurable via GPIO register.
	USB_C_PWR_EN2	O		**USB1 Port Power Enable**
	SPI1_CS4_N	O		**SPI1 Chip Select 4**
EA9	**IO03_CKO2_I2S1CLK**	Prg	Prg	**General Purpose I/O Function[P0.3]** This function is configurable via GPIO register. *Note: This pin reads in pin strapping information during reset for Boot 1 Boot Mode Strap IO (BOOT1 of BOOT[3..0])*
	CLKOUT2	O		**Clock Output 25 MHz** The pin is configurable to the clock with frequencies of 25 MHz.
	I2S_CLK1	I/O		**I2S Interface 1 Clock**
CL8	**IO04_LEDST_LEDDD0_SPISLVCK**	I	Prg	**SPI Slave Clock (function not available in product)**
	LED_ST	O		**LED Strobe** This is the store signal for the external shift register. It is either a clock or a pulse. Configuration is done in the LED controller register LED_CONx.
	LED_DD0	O		**Hardware Indication LED Pin 0**
	GPIO4	Prg		**General Purpose I/O Function[P0.4]** This function is configurable via GPIO register.
CJ4	**IO05_LEDD_LEDDD1_SPISLVI**	I	Prg	**SPI Slave Receive (function not available in product)**
	LED_D	O		**LED Data** This is the serial data output for the external shift register.
	LED_DD1	O		**Hardware Indication LED Pin 1**
	GPIO5	Prg		**General Purpose I/O Function[P0.5]** This function is configurable via GPIO register.

Table 22 GPIO/Flash/I²C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
CM2	**IO06_LEDSH_LEDDD2_SPISLVO**	O	Prg	**SPI Slave Transmit (function not available in product)**
	LED_SH	O		**LED Shift Clock** This is the shift clock for the external shift register.
	LED_DD2	O		**Hardware Indication LED Pin 2**
	GPIO6	Prg		**General Purpose I/O Function[P0.6]** This function is configurable via GPIO register.
EL47	**IO07_LEDSH1_CKOUT1_FANIN**	Prg	Prg	**General Purpose I/O Function[P0.7]** This function is configurable via GPIO register.
	LED_SH1	O		**LED Shift Clock 1** This is the shift clock for the external shift register.
	CLKOUT1	O		**Clock Output 40 MHz**
	FAN_CTRL_I	I		**Fan Control Input**
BY26	**IO08_VSPI1CS1_ZSIFSC_CKO0**	Prg	Prg	**General Purpose I/O Function[P0.8]** This function is configurable via GPIO register.
	C55_SPI1_CS1	O		**C55 SPI1 Chip Select 1**
	CLKOUT0	O		**Clock Output 8.192 MHz** The pin frequency is configurable to 8.192 MHz.
CN5	**IO09_SPI2CS1**	I	Prg	**SPI Slave Chip Select (function not available in product)**
	PON_Tx_SD	I		**PON Tx SD**
	SPI2_CS1_N	O		**SPI2 Chip Select 1**
	GPIO9	Prg		**General Purpose I/O Function[P0.9]** This function is configurable via GPIO register.
DE5	**IO10_URT2TX_SPI1TX_SSI1TX**	Prg	Prg	**General Purpose I/O Function[P0.10]** This function is configurable via GPIO register.
	SPI1_TX	O		**SPI1 Transmit** This pin function is only available in SPI mode.
	SSI1_TX	O		**SmartSLIC1 Transmit**
	UART2_TX	O		**UART2 Transmit**
DG10	**IO11_URT2RX_SPI1RX_SSI1RX**	Prg	Prg	**General Purpose I/O Function[P0.11]** This function is configurable via GPIO register.
	SPI1_RX	I		**SPI 1 RX**
	SSI1_RX	I		**SmartSLIC1 Receive**
	UART2_RX	I		**UART2 Receive**
EA54	**IO12_I2C0SDA-PMIC**	I/O	Prg	**I2C #0 SDA**
	GPIO12	I/O		**General Purpose I/O [P0.12]**
	SPI0_CS6_N	O		**SPI 0 Chip Select 6**

Table 22 GPIO/Flash/I²C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
DL15	**IO13_NDALE_SPI1CS3**	O	Prg	**NAND Flash Address Latch Enable** The address latch enable loads an address into the target.
	GPIO13	Prg		**General Purpose I/O Function[P0.13]** This function is configurable via GPIO register.
	SPI1_CS3_N	O		**SPI 1 Chip Select 3**
DH6	**IO14_URT2RTS_SPI1CS0_VRST1**	Prg	Prg	**General Purpose I/O Function[P0.14]** This function is configurable via GPIO register. *Note: This pin reads in pin strapping information during reset for Boot 3 Boot Mode Strap IO (BOOT3 of BOOT[3..0])*
	SPI1_CS0_N	O		**SPI1 Chip Select 0** This pin function is available in SPI 1.
	SLIC_RST1	O		**SmartSLIC1 Interface Reset** This pin function is only available in SSI mode.
	UART2_RTS	O		**UART2 HW Flow Control Signal**
CV9	**IO15_SPI0CS1_SPI1CS0**	Prg	Prg	**General Purpose I/O Function[P0.15]** This function is configurable via GPIO register.
	SPI0_CS1_N	I/O		**SPI0 Chip Select 1** This pin output function is only available in SPI master mode. When it is slave, it is an input. This CS is used for serial flash boot.
	C55_SPI1_CS0	O		**C55 Voice SPI1 Chip Select 0**
DA11	**IO16_SPI0DI_SPI1RX**	Prg	Prg	**General Purpose I/O Function[P0.16]** This function is configurable via GPIO register.
	SPI0_DIN	I/O		**SPI0 Data Input** In slave mode, this pin is output. In master mode, this pin is input.
	C55_SPI1_RX	I		**C55 Voice SPI1 RX**
DB7	**IO17_SPI0DO_SPI1TX**	Prg	Prg	**General Purpose I/O Function[P0.17]** This function is configurable via GPIO register.
	SPI0_DOUT	I/O		**SPI0 Data Output** In slave mode, this pin is input. In master mode, this pin is output.
	C55_SPI1_TX	O		**C55 Voice SPI1 Tx**
DB4	**IO18_SPI0CK_SPI1CK**	Prg	Prg	**General Purpose I/O Function[P0.18]** This function is configurable via GPIO register.
	SPI0_CLK	I/O		**SPI0 Clock** In slave mode, the clock is input. In master mode, the clock is output and only active when data is transmitted.
	C55_SPI1_CLK	O		**C55 Voice SPI1 Clock**

Table 22 GPIO/Flash/I²C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
DE2	**IO19_URT2CTS_SPI1CK_SSI1CK**	Prg	Prg	**General Purpose I/O Function[P0.19]** This function is configurable via GPIO register. *Note: This pin reads in pin strapping information during reset for Boot 2 Boot Mode Strap IO (BOOT2 of BOOT[3..0])*
	SPI1_CLK	O		**SPI1 Clock** This pin function is available in SPI 1.
	SSI1_CLK	I		**SmartSLIC1 Interface Clock** This pin function is only available in SSI mode.
	UART2_CTS	O		**UART2 HW flow Control Signal**
DV52	**IO20_I2C0SCL-PMIC**	O	Prg	**I2C #0 SCL**
	GPIO20	I/O		**General Purpose I/O[P0.20]**
	SPI1_CS5_N	O		**SPI #1 Chip Select 5**
DY6	**IO21_I2C1SDA**	Prg	Prg	**General Purpose I/O Function[P0.21]** This function is configurable via GPIO register.
	I2C_SDA1	I/O		**I2C #1 Serial Data (SDA)**
DU2	**IO22_I2C1SCL**	Prg	Prg	**General Purpose I/O Function[P0.22]** This function is configurable via GPIO register.
	I2C_SCL1	O		**I2C #1 Serial Clock Line (SCL)**
EF51	**IO23_NDCS1_LEDST1**	O	Prg	**NAND Flash Chip Select 1** This NAND chip select is used by ROM for booting.
	GPIO23	Prg		**General Purpose I/O Function[P0.23]** This function is configurable via GPIO register.
	LED_ST1	O		**LED SSO 1 Strobe**
DF26	**IO24_NDCLE_SPI0CS4**	O	Prg	**NAND Flash Command Latch Enable**
	GPIO24	Prg		**General Purpose I/O Function[P0.24]** This function is configurable via GPIO register.
	SPI0_CS4_N	O		**SPI 0 Chip Select 4**
BY23	**IO25_NDBC0_1588GPC2**	O	Prg	**NAND BC0**
	GPIO25	I/O		**General Purpose I/O[P0.25]**
	GPC2_1588	I/O		**General Purpose Clock 2 for 1588**
BP19	**IO26_NDBC1**	O	Prg	**NAND BC1**
	GPIO26	I/O		**General Purpose I/O[P0.26]**
CG23	**IO27_SPI1CS1_VRST0**	I/O	Prg	**General Purpose I/O[P0.27]**
	SLIC_RST0	O		**SLIC Reset**
	SPI1_CS1_N	O		**SPI #1 Chip Select 1**
	C55_RESET0	O		**C55 SPI Reset 0**

Table 22 GPIO/Flash/I²C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
DR8	**IO28_ZSIFSC_TDM1FSC**	O	Prg	**General Purpose I/O[P0.28]** Supplied by 3.3 V only.
	TDM1_FSC	I/O		**TDM Frame Sync**
DM7	**IO29_ZSIM2S_TDM1DO**	O	Prg	**General Purpose I/O[P0.29]** Supplied by 3.3 V only.
	TDM1_DO	O		**TDM Data Output**
DM4	**IO30_ZSIS2M_TDM1DI**	O	Prg	**General Purpose I/O[P0.30]** Supplied by 3.3 V only.
	TDM1_DI	I		**TDM Data Input**
DK9	**IO31_ZSIDCL_TDM1DCL**	O	Prg	**General Purpose I/O[P0.31]** Supplied by 3.3 V only.
	TDM1_DCL	I/O		**TDM Data Clock**
DV33	**IO32_LEDD1_TDM0FSC_I2S1WA**	I/O	Prg	**General Purpose I/O**
	LED_D1			**LED Controller 1**
	I2S_WA1	O		**I2S1 WA**
	TDM0_FSC	O		**TDM FSC**
DL33	**IO33_TDM0DO_I2S1TX**	I/O	Prg	**General Purpose I/O**
	TDM0_DO	O		**TDM Data output**
	I2S_TX1	O		**I2S1 Tx**
BY19	**IO34_VSPI1TX_ZSIM2S_SSI0TX**	I/O	Prg	**General Purpose I/O**
	SSI0_TX	O		**SmartSLIC0 Interface Tx**
	C55_SPI1_TX	O		**Tx of SPI1 C55**
CD12	**IO35_VSPI1RX_ZSIS2M_SSI0RX**	I/O	Prg	**General Purpose I/O**
	SSI0_RX	I		**SmartSLIC0 Interface Rx**
	C55_SPI1_RX	I		**Rx of SPI1 of C55**
BY15	**IO36_VSPI1CK_ZSIDCL_SSI0CK**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	SSI0_CLK	I		**SmartSLIC0 Interface Clock** This pin function is only available in SSI mode.
	C55_SPI1_CLK	O		**CLK of SPI1 C55**

Table 22 GPIO/Flash/I²C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
DH35	**IO37_TDM0DI_I2S1RX**	I/O	Prg	**General Purpose I/O**
	TDM0_DI	I		**Data input of TDM0**
	I2S_RX1	I		**I2S1 Rx**
EA30	**IO38_TDM0DCL_I2S1CK**	I/O	Prg	**General Purpose I/O**
	TDM0_DCL	O		**DCL of TDM0**
	I2S_CLK1	O		**I2S Clock 1**
ED52	**IO39_SPI3CS1**	I/O	Prg	**General Purpose I/O**
	PON_LOS	O		**PON_Rx_LOS**
	SPI3_CS1_N	O		**CS1 of SPI3**
DV28	**IO40**	I/O	Prg	**General Purpose I/O**
	LED_DD3			**Direct LED DATA 3**
CH11	**IO41_MDIO0CKWAN_FANIN**	I/O	Prg	**General Purpose I/O**
	MDIO0_WANC	O		**WAN MDIO Clock**
	FAN_CTRL_IN	I		**Fan Control Input**
	LED_DD4			**Direct LED DATA 4**
CH7	**IO42_MDIO0DWAN_FANOUT**	I/O	Prg	**General Purpose I/O**
	MDIO0_WAN	I/O		**WAN MDIO Data**
	LED_DD5			**Direct LED DATA 5**
	FAN_CTRL_OUT	O		**FAN Controller**
CY35	**IO43**	I/O	Prg	**General Purpose I/O**
	SATA_MP_Switch	I/O		**SATA Mechanical Port Switch Control**
DH31	**IO44_I2S0CK_SPI2CK**	I/O	Prg	**General Purpose I/O**
	I2S_CLK0	I/O		**I2S Clock 0**
	SPI2_CLK	O		**SPI2 Clock**
DD35	**IO45_I2S0WA_SPI2TX**	I/O	Prg	**General Purpose I/O**
	I2S_WA0	I/O		**I2S WA0**
	SPI2_TX	O		**SPI2 Transmit**
DP28	**IO46_I2S0RX_SPI2RX**	I/O	Prg	**General Purpose I/O**
	I2S_RX0	I/O		**I2S RX 0**
	SPI2_RX	I		**SPI2 Receive**
DL28	**IO47_I2S0TX_SPI2CS0**	I/O	Prg	**General Purpose I/O**
	I2S_TX0	I/O		**I2S TX0**
	SPI2_CS0_N	O		**SPI2 Chip Select 0**

Table 22 GPIO/Flash/I²C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
DF15	**IO48_NDRDBYN**	I	Prg	**NAND Flash Ready /Busy** The ready/busy signal indicates the NAND device status. When low, the signal indicates that no more NAND operations are in progress.
	GPIO48	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
DF19	**IO49_NDRDN**	O	Prg	**NAND Flash Read Enable** The NAND read enable controls the latching of input data.
	GPIO49	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
CW23	**IO50_NDD1_QSPID1**	I/O	Prg	**NAND Flash Command/Address/Data Pin 1**
	GPIO50	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	QSPI_D1	I/O		**Data 1 of QSPI**
CW26	**IO51_NDD0_QSPID0**	I/O	Prg	**NAND Flash Command/Address/Data Pin 0**
	GPIO51	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	QSPI_D0	I/O		**Data 0 of QSPI**
CW19	**IO52_NDD2_QSPID2**	I/O	Prg	**NAND Flash Command/Address/Data Pin 2**
	GPIO52	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	QSPI_D2	I/O		**Data 2 of QSPI**
CN15	**IO53_NDD7_QSPID3**	I/O	Prg	**NAND Flash Command/Address/Data Pin 7**
	GPIO53	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	QSPI_D3	I/O		**Quad SPI Data 3**
CN19	**IO54_NDD6_QSPICK**	I/O	Prg	**NAND Flash Command/Address/Data Pin 6**
	GPIO54	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	QSPI_CLK	I/O		**QSPI Clock** In master mode, the clock is output and only active when data is transmitted.
CN26	**IO55_NDD5_QSPIRST0**	I/O	Prg	**NAND Flash Command/Address/Data Pin 5**
	GPIO55	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	QSPI_RST0	O		**Reset of QSPI 0**
CN23	**IO56_NDD4_QSPICS0**	I/O	Prg	**NAND Flash Command/Address/Data Pin 4**
	GPIO56	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	QSPI_CS0	O		**QSPI Chip Select 0**

Table 22 GPIO/Flash/I²C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
CW15	**IO57_NDD3_QSPICS1**	O	Prg	**NAND Flash Command/Address/Data Pin 3**
	GPIO57	Prg		**General Purpose I/O**
	QSPI_CS1	O		**QSPI Chip Select 1**
DF23	**IO58_NDCS0_QSPIRST1**	O	Prg	**NAND Flash Chip Select 0** This is NAND chip select 0.
	GPIO58	I/O		**General Purpose I/O Function** This function is configurable via GPIO register.
	QSPI_RST1	O		**QSPI Reset 1**
DM11	**IO59_NDWRN**	O	Prg	**NAND Flash Write Enable** The NAND write enable control the sending of output data.
	GPIO59	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
DW11	**IO60_NDWPN_VINT1_I2STX1**	O	Prg	**NAND Flash Write Protect** The write protect signal disables flash array program and erase operations.
	GPIO60	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	SLIC2C55_INT1			**Interrupt 1 to C55 SLIC2**
	I2S_TX1	I/O		**I2S TX1**
EH46	**IO61_NDSE_VINT0_I2SWA1**	O	Prg	**NAND Flash Spare Area Enable** The spare area enable output controls the access of the spare area. When SE is high, the spare area is not accessible for reading or programming. This is used in legacy NAND devices.
	GPIO61	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	SLIC2C55_INT0	I		**Interrupt 0 to C55 SLIC2**
	I2S_WA1	I/O		**I2S WA1**
EE1	**IO62_I2C3SCL_I2SRX1**	I/O	Prg	**General Purpose I/O**
	I2C_3_SCL	O		**I2C_3 SCL**
	I2S_RX1	I		**I2S RX1**
DV22	**IO63_I2C3SDA**	I/O	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	I2C_3_SDA	I/O		**I2C_3 SDA**
CP10	**IO64_URT0RX**	I	Prg	**UART0 RX**
	GPIO64	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
CR6	**IO65_URT0TX**	O	Prg	**UART0 Tx**
	GPIO65	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.

Table 22 GPIO/Flash/I²C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
EF14	**IO66_URT1RX_TDM2FSC**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	UART1_RX	I		**UART1 Rx**
	C55_TDM2_FSC	O		**FSC of TDM2 C55**
	TDM2_FSC	O		**TDM 2 Frame Sync Clock**
EJ13	**IO67_URT1TX_TDM2DO**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register. *Note: This pin reads in pin strapping information during reset for Boot 0 Boot Mode Strap IO (BOOT0 of BOOT[3..0])*
	UART1_TX	O		**UART1 Tx**
	C55_TDM2_DO	O		**Data output of TDM2 C55**
	TDM2_DO	O		**Data output of TDM2**
EK3	**IO68_URT2RX_TDM2DI**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	UART2_RX	I		**UART2 Rx**
	C55_TDM2_DI	I		**Data in of TDM2 C55**
	TDM2_DI	I		**TDM 2 Data In**
EH63	**IO69_URT2TX_TDM2DCL**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	UART2_TX	O		**UART2 Tx**
	C55_TDM2_DCL	O		**DCL of TDM2 C55**
	TDM2_DCL	O		**TDM 2 Data Clock**
EE60	**IO70_I2C2DA_URT3RX_SPI3CS0**	I	Prg	**UART3 Rx**
	I2C_SDA2	I/O		**SDA of I2C 2**
	GPIO70	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	SPI3_CS0_N	O		**SPI3 Chip Select 0**
EE62	**IO71_I2C2CK_URT3TX_SPI3TX**	O	Prg	**UART3 Tx**
	I2C_SCL2	O		**SCL of I2C 2**
	GPIO71	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	SPI3_TX	O		**SPI3 Transmit**
EH59	**IO72_URT3RTS_SPI3RX**	I	Prg	**UART3 RTS**
	GPIO72	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	TX_Fault	I		**PON Tx_Fault**
	SPI3_RX	I		**SPI3 Data IN (Receive)**

Table 22 GPIO/Flash/I²C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
EL59	**IO73_URT3CTS_SPI3CK**	O	Prg	**UART3 CTS**
	GPIO73	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	TX_DISABLE	O		**PON TX Disable**
	SPI3_CLK	O		**SPI3 Clock**
DV40	**IO74_NDD13_SDCK**	O	Prg	**NAND Flash Command/Address/Data Pin 13**
	GPIO74	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	SDIO_CLK	O		**SDIO Clock**
	SATA0_PHY_DEVSLP	O		**SATA 0 PHY Dev Sleep** Places the PHY into device sleep power management state when asserted
DP40	**IO75_NDD12_SDCMD**	O	Prg	**NAND Flash Command/Address/Data Pin 12**
	GPIO75	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	SDIO_CMD	O		**SDIO Command**
	SATA1_PHY_DEVSLP	O		**SATA 1 PHY Dev Sleep** Places the PHY into device sleep power management state when asserted
DP33	**IO76_NDD10_SDD2**	O	Prg	**NAND Flash Command/Address/Data Pin 10**
	GPIO76	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	SATA2_MP_SWH	I		**SATA 2 Mechanical Presence switch** State of the external device presence switch. 0, closed; 1, open
	SDIO_D2	I/O		**SDIO Data 2**
DL40	**IO77_NDD11_SDD3**	O	Prg	**NAND Flash Command/Address/Data Pin 11**
	GPIO77	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	SATA2_CP_DET	I		**SATA Cold Presence Detection** Detects addition or removal of the powered-down device. 0, device removed; 1, device added
	SDIO_D3	I/O		**SDIO Data 3**
DP46	**IO78_NDD9_SDD1**	O	Prg	**NAND Flash Command/Address/Data Pin 9**
	GPIO78	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	SATA3_MP_SWH	I		**SATA 3 MP Switch** State of the external device presence switch. 0, closed; 1, open
	SDIO_D1	I/O		**SDIO Data 1**

Table 22 GPIO/Flash/I²C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
EA42	**IO79_NDD8_SDD0**	O	Prg	**NAND Flash Command/Address/Data Pin 8**
	GPIO79	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	SATA3_CP_DET	I		**SATA3**
	SDIO_D0	I/O		**SDIO Data 0**
DV46	**IO80_NDD14_SDCD**	I/O	Prg	**NAND Flash Command/Address/Data Pin 14**
	GPIO80	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	SATA1_MP_SWH	I		**SATA1** the state of the external device presence switch. 0, closed; 1, open
	SDIO_CD	I		**SDIO Card Detect**
EN36	**IO81_NDD15_SDWP**	I/O	Prg	**NAND Flash Command/Address/Data Pin 15**
	GPIO81	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
	SATA1_CP_DET	I		**SATA1** Detects addition or removal of the powered-down device. 0, device removed; 1, device added
	SDIO_WP	O		**SDIO Write Pulse** This pin function is only available in SDIO mode.
DL19	**IO82_LEDBD**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	SATA0_CP_DET	I		**SATA0 CP Detect** Detects addition or removal of the powered-down device. 0, device removed; 1, device added
	LED_BD	I/O		**Brightness Detection Input and Control** Indicates the brightness information and controls the external detector.
EE48	**IO83**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	PCI10_CLK_REQ	O		**CLK Req of PCIe10**
EG9	**IO84**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	PCI11_CLK_REQ	I		**CLK Req of PCIe11**
EN8	**IO85**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	PCI20_CLK_REQ	I		**CLK Req of PCIe20**
EK9	**IO86**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	PCI21_CLK_REQ	I		**CLK Req of PCIe21**

Table 22 GPIO/Flash/I²C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
DL23	**IO87**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	PCI30_CLK_REQ	I		**CLK Req of PCIe30**
EJ50	**IO88**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	PCI31_CLK_REQ	I		**CLK Req of PCIe31**
DT5	**IO89**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	PCI40_CLK_REQ	I		**CLK Req of PCIe40**
EJ37	**IO90_PMICIO5**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	PMIC_GPIO5	I		**GPIO5 for ext PMIC** Input from PMIC to indicate PMIC interrupt events
	PCI41_CLK_REQ	I		**CLK Req of PCIe41**
BP15	**IO91_PPSI-O_LEDDD6**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	PPS1_IN_OUT	I/O		**PPS 1 Input/Output**
	LED_DD6	O		**LED DD6**
DD8	**IO92_PPSO-I_LEDDD7**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	PPS2_IN_OUT	I/O		**PPS 2 Output/Input**
	LED_DD7	O		**LED DD7**
EL63	**IO93_CK32K_NTR_LEDDD9**	Prg	PrgPrg	**General Purpose I/O Function** This function is configurable via GPIO register.
	CLK32K	O		**32 kHz Input**
	NTR			**NTR Clock Input**
	LED_DD9	I/O		**LED DD9**
BB35	**IO94_MDIO1CKLAN**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	MDIO1_CLK	O		**LAN MDIO 1 Clock** This function is configurable via GPIO register. Used as MDIO master to access Ethernet subsystem PHYs.
BP26	**IO95_MDIO1DLAN**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	MDIO1_D	I/O		**LAN MDIO 1 Data** This function is configurable via GPIO register. Used as MDIO master to access Ethernet subsystem PHYs.

Table 22 GPIO/Flash/I²C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
EB38	**IO96_PMICIO0**	I	Prg	**GPIO 0 to ext PMIC** Input from PMIC to indicate power good
	GPIO96	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
EB27	**IO97_PMICIO1**	O	Prg	**GPIO 1 to ext PMIC** Output to PMIC to indicate power events request
	GPIO97	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
EC32	**IO98_PMICIO2**	I	Prg	**GPIO 2 to ext PMIC** Input from PMIC to indicate CPU0 rail state
	GPIO98	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
BB31	**IO99_LEDDD8**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	LED_DD8	O		**LED Direct Pin 8**
	Sata2_PHY_devslp	O		**SATA 2 PHY Dev Sleep**
CG19	**IO100_VSCC1RST1**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	C55_SCC1_RESET1	O		**SPI1 Reset of C55**
CG15	**IO101_VSPI0_CS1**	Prg	Prg	**General Purpose I/O Function** This function is configurable via GPIO register.
	C55_SPI0_CS1	O		**SPI0 cs1 of C55**
	Sata3_PHY_devslp	O		**SATA 3 PHY Dev Sleep**
EL34	**IO102_PMICIO3**		Prg	**GPIO3 to ext PMIC** Input from PMIC to indicate CPU1 rail state
	GPIO102	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.
EE35	**IO103_PMICIO4**	I	Prg	**GPIO4 to ext PMIC** Input from PMIC to indicate ADP rail state
	GPIO103	Prg		**General Purpose I/O Function** This function is configurable via GPIO register.

2.3.14 eMMC

The eMMc interface is intended as a flash interface.

Table 23 eMMC Interface

Ball No.	Name	Pin Type	Buffer Type	Function
EH22	**EMMC_D0**	I/0	A	**emmc_D0** Data pin
EE24	**EMMC_D1**	I/0	A	**emmc_D1** Data pin
EN25	**EMMC_D2**	I/0	A	**emmc_D2** Data pin
EC20	**EMMC_D3**	I/0	A	**emmc_D3** Data pin
EL21	**EMMC_D4**	I/0	A	**emmc_D4** Data pin
EA17	**EMMC_D5**	I/0	A	**emmc_D5** Data pin
EJ26	**EMMC_D6**	I/0	A	**emmc_D6** Data pin
EF26	**EMMC_D7**	I/0	A	**emmc_D7** Data pin
EG18	**EMMC_STRB**	I/0	A	**emmc_Strobe** Control pin
ED16	**EMMC_CMD**	I/0	A	**emmc_cmd** Control pin
EG31	**CALPAD**	I/0	A	**emmc_calibration** 10k resistor to ground
EB14	**EMMC_CLK**	I/0	A	**emmc_clk**
ED29	**vCTL_TP**	I/0	A	**emmc test point**

2.3.15 JTAG

Table 24 describes the JTAG interface.

Table 24 JTAG Signals

Ball No.	Name	Pin Type	Buffer Type	Function
DD31	**J_TRST**	I	PU	**JTAG Reset** *Note:* *1. Active low.* *2. When the JTAG interface is not in use, this pin must be connected to V_{SS}.* *3. A low-to-high transition of the TRST pin latches the JTAG mode (depending on the boot strap setting of pin J_TDO).*
	J_TRST	I	PU	**JTAG Reset**
CT35	**J_TDI**	I	PU	**Test Data Input**
	J_TDI	I		**JTAG Test Data Input**
CY31	**J_TDO**	O	–	**Test Data Output** There is a boot strap setting on this pin. This pin is used to decide the interface. 1 = JTAG debugging mode; 0= JTAG test mode
	J_TDO	O		**JTAG Test Data Output**
CT31	**J_TMS**	I	–	**Test Mode Select**
	J_TMS	I		**JTAG Mode Select**
DP22	**J_TCK**	I	–	**Test Clock**
	J_TCK	I		**JTAG Test Clock**

2.3.16 DDR-SDRAM

Table 25 describes the Double Data Rate (DDR) SDRAM interface.

DDR ball names, excluding Power, start with DDR_. When the function is unique for LPDDR4, LPDDR3, DDR4, or DDR3, only one name is shown after DDR_. When there is a different function for at least one DDR type, the convention after DDR_ is: LPDDR4_LPDDR3_DDR4_DDR3.

Attention: LPDDR3 support is removed from URX851.

Table 25 DDR-SDRAM Signals

Ball No.	Name	Pin Type	Buffer Type	Function
DM118	**DDR_DQ31**	I/O	–	**DDR Data Bus [31:16]**
DM111	**DDR_DQ30**	I/O	–	
DK113	**DDR_DQ29**	I/O	–	
ED118	**DDR_DQ28**	I/O	–	
EC115	**DDR_DQ27**	I/O	–	
EA113	**DDR_DQ26**	I/O	–	
DY116	**DDR_DQ25**	I/O	–	
DU120	**DDR_DQ24**	I/O	–	
DE120	**DDR_DQ23**	I/O	–	
DG112	**DDR_DQ22**	I/O	–	
DH116	**DDR_DQ21**	I/O	–	
CP112	**DDR_DQ20**	I/O	–	
CR116	**DDR_DQ19**	I/O	–	
CV113	**DDR_DQ18**	I/O	–	
DA111	**DDR_DQ17**	I/O	–	
DB115	**DDR_DQ16**	I/O	–	

Table 25 DDR-SDRAM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
EC103	**DDR_DQ15**	I/O	–	**DDR Data Bus [15:0]**
EH100	**DDR_DQ14**	I/O	–	
EE99	**DDR_DQ13**	I/O	–	
EK113	**DDR_DQ12**	I/O	–	
EG113	**DDR_DQ11**	I/O	–	
EJ109	**DDR_DQ10**	I/O	–	
EF108	**DDR_DQ9**	I/O	–	
EB108	**DDR_DQ8**	I/O	–	
ED94	**DDR_DQ7**	I/O	–	
EF97	**DDR_DQ6**	I/O	–	
EJ97	**DDR_DQ5**	I/O	–	
EB84	**DDR_DQ4**	I/O	–	
EF84	**DDR_DQ3**	I/O	–	
EJ85	**DDR_DQ2**	I/O	–	
EE87	**DDR_DQ1**	I/O	–	
EH89	**DDR_DQ0**	I/O	–	
EH76	**DDR_ECCDQS_P**	I/O	–	**Differential DQS Pair for ECC DQ lines [EDQ0 - EDQ7]**
EC77	**DDR_ECCDQS_N**	I/O	–	
EE74	**DDR_ECC_DM**	I/O	–	**ECC Byte Mask**
EB71	**DDR_ECCDQ7**	I/O	–	**ECC Bit [7:0]**
EJ72	**DDR_ECCDQ6**	I/O	–	
EF72	**DDR_ECCDQ5**	I/O	–	
EL75	**DDR_ECCDQ4**	I/O	–	
ED70	**DDR_ECCDQ3**	I/O	–	
EG78	**DDR_ECCDQ2**	I/O	–	
ED81	**DDR_ECCDQ1**	I/O	–	
EG67	**DDR_ECCDQ0**	I/O	–	
DL76	**DDR_NC_NC_ACTN_A15**	O	–	**A15 in DDR4 mode only** DDR CMD ACTIVATE, shared with RAS_N in DDR3 mode

Table 25 DDR-SDRAM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
DV90	**DDR_CKAC_CKAC_BG1_A14**	O	–	**DDR Address Bus [14:0]**
DV101	**DDR_CAB0_CAB0_A13_A13**	O	–	
CN107	**DDR_CAA0_CAA0_A12_A12**	O	–	
CW104	**DDR_CAA1_CAA1_A11_A11**	O	–	
DP95	**DDR_CAB2_CAB2_A10_A10**	O	–	
DP64	**DDR_NC_NC_A9_A9**	O	–	
DP76	**DDR_CAA3_CAA3_A8_A8**	O	–	
DL82	**DDR_CAA2_CAA2_A7_A7**	O	–	
DV76	**DDR_CAA4_CAA4_A6_A6**	O	–	
DP82	**DDR_CAA5_CAA5_A5_A5**	O	–	
DP58	**DDR_NC_CAA6_A4_A4**	O	–	
DV70	**DDR_NC_CAA7_A3_A3**	O	–	
CW97	**DDR_CKBT_CKBT_A2_A2**	O	–	
CW100	**DDR_CKBC_CKBC_A1_A1**	O	–	
DV95	**DDR_CAB3_CAB3_A0_A0**	O	–	
CG107	**DDR_NC_CAB6_WEN_WEN**	O	–	**DDR Write Enable; DDR4 A14**
BY100	**DDR_NC_CAB7_RASN_RASN**	O	–	**DDR Row Address Strobe;DDR4 A16**
DH83	**DDR_CAB5_CAB5_CASN_CASN**	O	–	**DDR Column Address Strobe;DDR4 A15**
CN100	**DDR_NC_CAB8_ODT0_ODT0**	O	–	**DDR UDM, LDM, ODT in DDR4 modes, DM[3:0] are shared with DBI_n[3:0]**
EL102	**DDR_DM1**	I/O	–	
EB96	**DDR_DM0**	I/O	–	
BP107	**DDR_NC_CAB9_ODT1_ODT1**	O	–	
DE117	**DDR_DM2**	I/O	–	
DM115	**DDR_DM3**	I/O	–	
DR114	**DDR_DQS3P**	I/O	–	**Differential DQS Pair for upper DQ lines [DQ24 - DQ31]**
DT117	**DDR_DQS3N**	I/O	–	
DD114	**DDR_DQS2P**	I/O	–	**Differential DQS Pair for lower DQ lines [DQ16- DQ23]**
DB118	**DDR_DQS2N**	I/O	–	
EG105	**DDR_DQS1P**	I/O	–	**Differential DQS Pair for upper DQ lines [DQ8 - DQ15]**
ED107	**DDR_DQS1N**	I/O	–	
EC91	**DDR_DQS0P**	I/O	–	**Differential DQS Pair for lower DQ lines [DQ0 - DQ7]**
EG92	**DDR_DQS0N**	I/O	–	

Table 25 DDR-SDRAM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
DP90	**DDR_CKAT_CKAT_BG0_BA2**	O	–	**DDR Bank/Group Address [1:0]; BG[0] is BA[2] for DDR3**
DP101	**DDR_CAB1_CAB1_BA0_BA0**	O	-	**DDR BANK**
CG100	**DDR_NC_NC_BA1_BA1**	O	-	**DDR BANK**
DF97	**DDR_CSA0_CSA0_CSN0_CSN0**	O	–	**LPDDR Ch A CS0/DDR Chip Select 0**
BY107	**DDR_NC_ODTB_CSN1_CSN1**	O	–	**LPDDR3 ODTB/DDR Chip Select 1**
DL70	**DDR_NC_CAA8_CK0P_CK0P**	O	–	**DDR Clock 0 Output**
DV64	**DDR_NC_CAA9_CK0N_CK0N**	O	–	**DDR Clock 0 Inverted**
BY104	**DDR_CKEB0_CKEB0_CK1P_CK1P**	O	–	**DDR Clock 1 Output**
BP104	**DDR_CKEB1_CKEB1_CK1N_CK1N**	O	–	**DDR Clock 1 Inverted**
DF100	**DDR_CKEA0_CKEA0_CKE0_CKE0**	O	–	**DDR Clock Enable 0**
CG97	**DDR_ZQ**	AI	–	**DDR Calibration Resistor, 120 Ω, 1%**
DL95	**DDR_RST**	O	–	**DDR RESET**
CN104	**DDR_NC_NC_CAPAR_CAPAR**	O	–	**DDR4 only; DDR Parity**
DL58	**DDR_NC_NC_ALTN_NC**	O	–	**DDR4 only; DDR4 ALERT_N**
DF104	**DDR_CKEA1_CKEA1_CKE1_CKE1**	O	–	**DDR Clock Enable 1**
CG104	**NC**	O	–	**Not Connected**
DL64	**DDR_NC_ODTA_NC_NC**	O	–	**ODT, not used in the system board**
CW107	**DDR_CSA1_CSA1_C0_NC**	O	–	**LPDDR Ch A CS1/DDR logic Channel Chip Select 0**
DL104	**DDR_CSB0_CSB0_NC_NC**	O	–	**LPDDR Ch B Chip Select 0**
DL100	**DDR_CSB1_CSB1_NC_NC**	O	–	**LPDDR Ch B Chip Select 1**
DL90	**DDR_CAB4_CAB4_NC_NC**	O	–	**LPDDR Command and Address Bus bit 4, Channel B**
BP100	**DDR_NC_NC_C1_NC**	O	–	**DDR4 logic channel Chip Select 1**
CN97	**DDR_NC_NC_MTEST_MTEST**	AIO	–	**DDR NC MTEST**

2.3.17 Clock

Table 26 describes the system clock signals.

Table 26 Clock Signals

Ball No.	Name	Pin Type	Buffer Type	Function
CC6	**XI**	AI	A	**External Crystal Input/Output**
CB10	**XO**	AO	A	
BN39	**CLK_1588_N**	AO	A	**CLK 1588 Differential Reference Clock** These pins provide the 25/40 MHz differential reference clock to be provided for clock slave device, connected to XO_ETH and XI_ETH.
BN43	**CLK_1588_P**	AO	A	
AA115	**PCIE10_CLKP**	AO	A	**PCIE10 Differential Reference Clock for RC Mode** These pins provide the 100 MHz differential reference clock to be provided for PCIe EP.
AA118	**PCIE10_CLKN**	AO	A	
AN118	**PCIE11_CLKP**	AO	A	**PCIE11 Differential Reference Clock for RC Mode** These pins provide the 100 MHz differential reference clock to be provided for PCIe EP.
AP116	**PCIE11_CLKN**	AO	A	
AA7	**PCIE20_CLKP**	AO	A	**PCIE20 Differential Reference Clock for RC Mode** These pins provide the 100 MHz differential reference clock to be provided for PCIe EP.
AA4	**PCIE20_CLKN**	AO	A	
AP7	**PCIE21_CLKP**	AO	A	**PCIE21 Differential Reference Clock for RC Mode** These pins provide the 100 MHz differential reference clock to be provided for PCIe EP.
AN4	**PCIE21_CLKN**	AO	A	
BW117	**PCIE30_CLKP**	AO	A	**PCIE30 Differential Reference Clock for RC Mode** These pins provide the 100 MHz differential reference clock to be provided for PCIe EP.
BW120	**PCIE30_CLKN**	AO	A	
BG111	**PCIE31_CLKP**	AO	A	**PCIE31 Differential Reference Clock for RC Mode** These pins provide the 100 MHz differential reference clock to be provided for PCIe EP.
BF115	**PCIE31_CLKN**	AO	A	
BW5	**PCIE40_CLKP**	AO	A	**PCIE40 Differential Reference Clock for RC Mode** These pins provide the 100 MHz differential reference clock to be provided for PCIe EP.
BW2	**PCIE40_CLKN**	AO	A	
BF4	**PCIE41_CLKP**	AO	A	**PCIE41 Differential Reference Clock for RC Mode** These pins provide the 100 MHz differential reference clock to be provided for PCIe EP.
BF7	**PCIE41_CLKN**	AO	A	

2.3.18 Reset

Table 27 describes the reset signals.

Table 27 Reset Signals

Ball No.	Name	Pin Type	Buffer Type	Function
EH34	**POR_N**	I	PU	**Power-on Reset** *Note: Active low*
CU12	**HRST_N**	O	OD	**Hardware Reset/Power-on Reset Output** This pin is the POR module output and synchronous with the on-chip reset after power-on. *Note: Active low*
AR31	**HRSTN_ETH**	I	PU	**Hardware Reset Input** This pin is the hardware reset for the Ethernet PHY subsystem, connect to HRST_N on the PCB. *Note: Active low*

2.3.19 Power

Chapter 15 Electrical Characteristics of URX851/URX850/MxL25641 describes the power supply characteristics.

Table 28 Power Supply

Ball No.	Name	Pin Type	Buffer Type	Function
XO POR and PLL Supply				
CE35	**VDDA1V8PORXO**	PWR	–	**1.8 V Power for XO, POR**
DH92	**VDDA1V8DDRPLL**	PWR	–	**1.8 V Power for DDR PLL**
CH46, CH51	**VDDA1V8ROPLL**	PWR	–	**1.8 V Power for PLL0/1/2**
CN46	**VDDA1V8CPUPLL**	PWR	–	**1.8 V Power for CPU PLL**
CA39, CA43	**VDDA1V8LCPLL**	PWR	–	**1.8 V Power for Low Jitter PLL**
CE39, CE43	**VDDD0V8CPUPLLPOST**	PWR	–	**0.80 V Digital Power** PLL post divider power for CPU module.
CH39, CH43	**VDDD0V8CPUPLLREF**	PWR	–	**0.80 V Digital Power** PLL REF Circuit power for CPU module.
BU31, BU35	**VDDD0V8LCPLLPOST**	PWR	–	**0.80 V Digital Power** Low jitter PLL post divider power supply.
BN31, BN35	**VDDD0V8LCPLLREF**	PWR	–	**0.80 V Digital Power** Low jitter PLL REF power supply.
ROC, ADP, JTAG and CPU Digital Supply				
AF79, AF83, AR65, AR69, AR76, AR79, AR83, AR88, AW79, AW88, BB69, BB72, BB76, BB83, BF79, BF88, BJ72, BJ76, BJ83, BN46, BN51, BN53, BN79, BN88, BN92, BU46, BU72, BU76, BU83, BU92, CA57, CA76, CA83, CE72, CN43, CT46, DL52, DP52	**VDDD0V8ROC**	PWR	–	**0.80 V Digital Power RoC Domain**
BB57, BB61, BB65, BF65, BF69, BJ57, BJ61, BJ65, BN61, BN69, BU57, BU61, BU65, CA69	**VDDD0V8ADP**	PWR	–	**0.8 V Power for ADP Domain Core Supply**
CH53, CH57, CH61, CH65, CN51, CN53, CN57, CN65, CT53, CY51, CY57, DD51, DD53, DD57	**VDDD0V8CPU0L2**	PWR	–	**0.80 V Digital Power** L2 and CPU module 0 domain power supply.

Table 28 Power Supply (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
CN69, CN72, CT61, CT69, CT72, CY61, CY69, DD61, DD65, DD69, DD72, DH61, DH69, DH72	**VDDD0V8CPU1L2**	PWR	–	**0.80 V Digital Power** L2 and CPU module 1 domain power supply.
CE46	**DVDD1V8_JTAG**	PWR	–	**1.8 V Power JTAG**
HSIO SerDes (PCI Express, PON, XFI) Supply				
AH97, AT97	**VPH_1V8_PCIE10_11**	PWR	–	**1.8 V Power for PCIe Analog of Module 10_11**
AE12, AH15	**VPH_1V8_PCIE20_21**	PWR	–	**1.8 V Power for PCIe Analog of Module 20_21**
AC76, AC82	**VPH_1V8_PON**	PWR	–	**1.8 V Power PON SerDes Analog**
AT15	**VPH_1V8_XFI5**	PWR	–	**1.8 V Power LAN XFI SerDes Analog**
AH104, AH107, AT104, AT107	**VA_0V8_PCIE10_11**	PWR	–	**0.80 V Power for PCIe Analog of Module 10_11**
AH19, AH23, AT19, AT23	**VA_0V8_PCIE20_21**	PWR	–	**0.80 V Power for PCIe Analog of Module 20_21**
BB104, BB107, BH104, BH107	**VA_0V8_PCIE30_31**	PWR	–	**0.80 V Power for PCIe Analog of Module 30_31**
BB19, BB23, BH19, BH23	**VA_0V8_PCIE40_41**	PWR	–	**0.80 V Power for PCIe Analog of Module 40_41**
BB97, BH97	**VPH_1V8_PCIE30_31**	PWR	–	**1.8 V Power for PCIe Analog of Module 30_31**
BB15, BH15	**VPH_1V8_PCIE40_41**	PWR	–	**1.8 V Power for PCIe Analog of Module 40_41**
W82, W90	**VA_0V8_PON**	PWR	–	**0.80 V Power for PON Analog of Module SerDes 1**
BF31, BJ31	**VA_0V8_XFI5**	PWR	–	**0.80 V Power for XFI5 Analog**
BF92, CH35	**VDDA0V8CML**	PWR	–	**0.80 V Power for CML Analog Module of PCIe Ref Clk**
AW92, CE31	**VDDA1V8CML**	PWR	–	**1.8 V Power for CML Analog Module of PCIe Ref Clk**

Table 28 Power Supply (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
USB Supply				
AF92, AK92	**VPH_1V8_USB0**	PWR	–	**1.8 V Power USB0 SerDes Analog**
AF88, AK88	**VPH_1V8_USB1**	PWR	–	**1.8 V Power USB1 SerDes Analog**
AD104, AD107, M106	**VA0V8_USB0**	PWR	–	**0.8 V Analog Supply of USB0**
R101, R95, W101	**VA0V8_USB1**	PWR	–	**0.8 V Analog Supply of USB1**
AC90	**VDDA3v3_USB2.1**	PWR	–	**3.3 V Power for USB 2.0 PHY Analog of USB 1**
AD100	**VDDA3v3_USB2.0**	PWR	–	**3.3 V Power for USB 2.0 PHY Analog of USB 0**
DDR Supply				
CT88, CY83, DD88, DV82	**VDDA0V6DDR**	PWR	–	**0.6 V Power for DDR** Connect to VDDA1V1DDR rail.
BY97, CA88, CD110, CE83, CE92, CH79, CH88, CM120, CN83, CN92, CY92, DD79, DF107, DH76, DP70, EA106, EE121, EL89, EM121, EN117	**VDDA1V1DDR**	PWR	–	**Power for DDR LPDDR4/DDR4/DDR3L** Supplied with 1.1/1.2/1.35 V respectively for different DDR modes. 1.1 V for LPDDR4/4x 1.2 V for DDR4 1.35 V for DDR3L.
CA79, CE76, CH76, CN76, CN79, CT76, CT79, CY76, CY79	**VDDD0V8DDR**	PWR	–	**Power for DDR Core Voltage** Supplied with 0.9 V for DDR I/F frequency >=3733 MHz; otherwise supplied with 0.8 V.
DL107	**VDD_VREF**	PWR	–	**Power for DDR Ref Voltage**
DDQ eMMC Supply				
DH46, DL46	**VDDA1V8EMMCQ**	PWR	–	**1.8 V Analog Power for EMMC DQ**
DD46	**VDDA0V8VCEMMC**	PWR	–	**0.8 V Analog Power for EMMC DLL Supply**
VDDP GPIO Supply				
CT39, CT43, CY39, CY43, DD39, DD43, DH39, DH43	**VDDD3V3GPIO**	PWR	–	**3.3 V Power for GPIO; Supply to 3.3 V only in System**
CH31	**VDDP1V8_SDIO**	PWR	–	**Power for SDIO** 1.8 V or 3.3 V. When SDIO interface is not in use, connect this ball to 3.3 V.

Table 28 Power Supply (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
Sense				
EC45	**M0_SENSE_P**	PWR	–	**Voltage Sensing for CPU Module 0**
EG44	**M1_SENSE_P**	PWR	–	**Voltage Sensing for CPU Module 1**
ED41	**ADP_SENSE_P**	PWR	–	**Voltage Sensing for ADP**
EF38	**ROC_SENSE_P**	PWR	–	**Voltage Sensing for RoC**

2.3.20 Power / Ground

Chapter 15 Electrical Characteristics of URX851/URX850/MxL25641 describes the power / ground supply.

Table 29 Power Supply Ground Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Ground Supply				
A117, A120, A86, A98, AB11, AC95, AE110, AG10, AG112, AH100, AL114, AL8, AM1, AM121, AP11, AP111, AR92, AT100, AU113, AV110, B121, BA10, BB100, BB26, BB92, BD114, BD8, BE1, BE121, BU43, BF118, BF35, BG11, BH100, BH113, BH26, BJ35, BK12, BM10, BR117, BT8, BU1, BU121, BU39, BV114, BV8, C119, CA31, CA35, CA46, CB112, CK121, CL114, E121, EN49, EN61, G113, H87, H99, K103, K115, K77, K91, L108, L84, L96, P111, R82, R90, V114, V8, Y1, Y121	**VSSA**	GND	–	**Analog Ground**
BF39	VSSXO_0	GND	–	
BU43	VSSXO_1	GN	–	
AF69, AF72, AF76, AK65, AK69, AK72, AK76, AK79, AR72, AW57, AW61, AW65, AW69, AW72, AW76, AW83, BB79, BF57, BF61, BF72, BF76, BF83, BJ51, BJ53, BJ69, BJ79, BJ88, BJ92, BN57, BN65, BN72, BN76, BN83, BP97, BU51, BU53, BU69, BU79, BU88, CA51, CA61, CA65, CA72, CA92, CE57, CE61, CE65, CE69, CE79, CE88, CG26, CH111, CH69, CH72, CH83, CH92, CK1, CN31, CN117, CN35, CN39, CN61, CN88, CT51, CT57, CT65, CT83, CT92, CU110, CY46, CY53, CY65, CY72, CY88, DC1, DC121, DD76, DD83, DD92, DH51, DH53, DH57, DH65, DH79, DH88, DJ110, DN1, DN121, DV58, DW111, EA68, EA80, EA93, EB51, EC66, EH1, EH121, EK119, EM1, EN114, EN120, EN2, EN5, EN73, EN86, EN98	**VSSP**	GND	–	**Ground for PAD** Ground for digital core, I/O logic

2.3.21 Fuse

Table 30 describes how the fuse pins must be connected.

Table 30 Fuse Signals

Ball No.	Name	Pin Type	Buffer Type	Function
CE51, CE53, CA53	**VDDA1V8OTPVT**	I	–	**OTP and PVT Power, 1.8 V** Source to program OTP memory and PVT sensors; connect to 1.8 V digital power in system board.
CF9	**VDDA1V8FUSE**	I	–	**Fuse Power Supply, 1.8 V**
AR61	**OTPGND**	PWR	–	**OTP Ground** Connect to digital ground in system board.

2.3.22 Pins from Ethernet Subsystem

Table 31 describes how the Ethernet pins must be connected.

Table 31 Ethernet Subsystem Signals

Ball No.	Name	Pin Type	Buffer Type	Function
G9	**PHYLED1_0**	O	Prg	**Ethernet PHY0 LED1**
A8	**PHYLED2_0**	O	Prg	**Ethernet PHY0 LED2**
C9	**PHYLED3_0**	O	Prg	**Ethernet PHY0 LED3**
K7	**PHYLED1_1**	O	Prg	**Ethernet PHY1 LED1**
H1	**PHYLED2_1**	O	Prg	**Ethernet PHY1 LED2**
J4	**PHYLED3_1**	O	Prg	**Ethernet PHY1 LED3**
R52	**PHYLED1_2**	O	Prg	**Ethernet PHY2 LED1**
R40	**PHY_GPIO0**	I/O	Prg	**Ethernet GPIO0** Either selected as input or output mode. The output characteristic is either open drain or push-pull. *Note:* 1. *This pin reads in pin strapping information during reset.* 2. *Pin strap URX851 mode.*
	EXTINT0	O	Prg	**Ethernet External Interrupt 0 as Alternative Function**
	ETH_UTXD	O	Prg	**Ethernet UART TXD**
R46	**PHY_GPIO1**	I/O	Prg	**Ethernet GPIO1**
	EXTINT1	O	Prg	**Ethernet External Interrupt1**
	ETH_URXD	I	Prg	**Ethernet UART RXD**
L14	**PHYLED2_2**	O	Prg	**Ethernet PHY2 LED2**
P11	**PHYLED3_2**	O	Prg	**Ethernet PHY2 LED3**

Table 31 Ethernet Subsystem Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
M42	**PHY_GPIO2**	O	Prg	**Ethernet PHY_GPIO2** This pin reads in pin strapping information during reset. It is also used as VID0 for setting supply voltage for Ethernet (default 0.95V).
M54	**PHYLED1_3**	O	Prg	**Ethernet PHY3 LED1**
R58	**PHYLED2_3**	O	Prg	**Ethernet PHY3 LED2**
R64	**PHYLED3_3**	O	Prg	**Ethernet PHY3 LED3**
AC70	**GPC1**	I/O	Prg	**Ethernet GPC1 for SyncE and 1588** This pin reads in pin strapping information during reset for PHY_ADDR4.
R70	**GPC2**	I/O	Prg	**Ethernet GPC2 for SyncE and 1588** This pin reads in pin strapping information during reset for PHY_ADDR3.
R76	**GPC3**	I/O	Prg	**Ethernet GPC3 for SyncE and 1588** This pin reads in pin strapping information during reset for PHY_ADDR2.
W70	**CLKO**	AO	–	**Clock out from Ethernet, 25 MHz** This pin reads in pin strapping information during reset.VID1 for setting supply voltage for Ethernet (default 0.95V).
AW35	**MDC**	I	–	**Ethernet MDIO Slave Clock** Connect to IO94_MDIO1CKLAN on the PCB
AW31	**MDINT**	O	Prg	**Ethernet MDIO Interrupt** This pin reads in pin strapping information during reset. Connect to IO99_LEDDD8 on the PCB.
BP23	**MDIO**	I/O	Prg	**Ethernet MDIO Data** Connect to IO95_MDIO1DLAN on the PCB.
L51	**TRST_ETH**	I	-	**Reset to Ethernet JTAG Interface**
W76	**URESREF**	AI/AO	A	**USXGMII SerDes Reference Resistor**
BF51	**URXM**	AI	A	**USXGMII SerDes RX** Connect to XFI5_XPCS_TXP on the PCB.
BB51	**URXP**	AI	A	**USXGMII SerDes RX+** Connect to IXFI5_XPCS_TXN on the PCB.
BF43	**UTXM**	AO	A	**USXGMII SerDes TX-** Connect to IXFI5_XPCS_RXP on the PCB.
BB43	**UTXP**	AO	A	**USXGMII SerDes TX+** Connect to IXFI5_XPCS_RXN on the PCB.
BJ43	**XI_ETH**	I		**Ethernet Clock Supply Input** Connect to CLK_1588__P on the PCB.
BJ39	**XO_ETH**	O		**Ethernet Clock Supply Output** Connect to CLK_1588__N on the PCB.

Table 31 Ethernet Subsystem Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
Ethernet Supply				
AF53	**VDDA3V3CDB**	PWR	–	**Power for Ethernet CDB, XO, PLL Module, 3.3 V**
AC28, AC33, AC40, AC46, AC52, AD23, AH26, AT26	**VDDHA3V3**	PWR	–	**Analog Power for Ethernet TPI, 3.3 V**
M30, R28, R33, W28, W33, W40, W46, W52	**VDDLA0V9**	PWR	–	**Analog Power for Ethernet TPI, 0.95 V**
AR46, AK46, AK31, AF57, AK43, AF61, AK35, AK39, AR39, AR43, AR51, AR53, AR57	**VDD0V9CORE**	PWR	-	**Core 0.95 V Power for Ethernet**
AF31, AF35, AF39, AF43, AF46, AF51, AK51, AK53, AK57, AK61, AR35, AW39, AW43, AW46, AW51, AW53	**VSSD**	GND	-	**GND for Ethernet Digital Core**
AC58, W58	**VDD3V3PAD**	PWR	-	**VDD 3.3 V for Ethernet Digital Pads**
A2, A25, A36, A49, A5, A61, A73, AC64, B1, C3, E1, H24, H35, H48, H60, H62, H74, J52, K20, K32, K45, K66, L27, L38, L71, M68, R22, W64	**TPVSSA**	GND	–	**Ground**
AD15, AD19, W22	**UVPA0V9**	PWR	-	**Ethernet Analog 0.95 V for SerDes, CDB**
BB39	**UVPHA1V8**	PWR	-	**Ethernet Analog 1.8 V for SerDes**

2.3.23 Ethernet Subsystem Boot Strapping Definition

Table 32 provides Ethernet subsystem pin boot strapping information.

Table 32 Strapping Information for Ethernet Subsystem

Pin	Boot Strapping Definition
PHY_GPIO0	**DATA Interface Mode** This is to specify which type of data interface is used: 0_B **USXGMII** USXGMII mode (used in URX851) 1_B **SGMII** SGMII mode (not used in URX851)
PHY_GPIO2	**RJ45 Tap Configuration/VID 0** This is to specify the tap-up or tap-down configuration of the RJ45. Each slice has the same configuration. It is also set the core voltage level together with CLKO pin as VID1. 0_B **UP** Tap-up 1_B **DOWN** Tap-down
GPC1	**MDIO PHY Address: PS_PHY_MADDR[4]** This is to specify the MDIO address most significant bit 4. The lower 2 bits are hard-coded to the same value as SLICE_CTRL_PDI_ID.HW_ID.
GPC2	**MDIO PHY Address: PS_PHY_MADDR[3]** This is to specify the MDIO address most significant bit 3. The lower 2 bits are hard-coded to the same value as SLICE_CTRL_PDI_ID.HW_ID.
GPC3	**MDIO PHY Address: PS_PHY_MADDR[2]** This is to specify the MDIO address most significant bit 2. The lower 2 bits are hard-coded to the same value as SLICE_CTRL_PDI_ID.HW_ID.
CLKO	**LED Setup Configuration /VID1** This is to specify the LED default setup configuration. It is also set the core voltage level together with PHY_GPIO2 pin as VID0 0_B **Custom configuration** take from GPIO registers 1_B **Default** Default
MDINT	**MDIO Interrupt Polarity** This is to specify the polarity of the MDIO interrupt. For automatic configuration of the MDIO interrupt polarity, connect this configuration bit to the input of the MDIO interrupt pad. 0_B **HIGH MDIO** Interrupt is active high 1_B **LOW MDIO** Interrupt is active low

Table 33 Ethernet PHY Subsystem Low Voltage Setting

(Set by Ethernet PHY firmware)	VID1	VID0	Voltage Setting	Notes
Pin Name	CLKO	PHY_GPIO2	**VDD0V9CORE,** UVPA0V9, **VDDLA0V9**	
Logic Level	0	0	0.85 v	
	0	1	0.90 v	
	1	0	0.95 v	Default
	1	1	1.00 v	

2.3.24 Ethernet Media Interface

Table 34 describes how the Ethernet interface pins must be connected.

Table 34 Ethernet Media Interface Signals

Pin No.	Name	Pin Type	Buffer Type	Function
Ethernet Port 0 Ethernet Media Interface				
E76	**TPIAPF0**	AI/AO	A	**Port 1 Transmit/Receive Positive/Negative** Connect directly to XFMR without any pull-down terminators, such as resistors or capacitors, required.
B75	**TPIANF0**	AI/AO	A	
D72	**TPIBPF0**	AI/AO	A	
G72	**TPIBNF0**	AI/AO	A	
J70	**TPICPF0**	AI/AO	A	
F67	**TPICNF0**	AI/AO	A	
E63	**TPIDPF0**	AI/AO	A	
B63	**TPIDNF0**	AI/AO	A	
B59	**TPIAPF1**	AI/AO	A	**Port 2 Transmit/Receive Positive/Negative** Connect directly to XFMR without any pull-down terminators, such as resistors or capacitors, required.
E59	**TPIANF1**	AI/AO	A	
F55	**TPIBPF1**	AI/AO	A	
K56	**TPIBNF1**	AI/AO	A	
G51	**TPICPF1**	AI/AO	A	
D50	**TPICNF1**	AI/AO	A	
B47	**TPIDPF1**	AI/AO	A	
E46	**TPIDNF1**	AI/AO	A	
F44	**TPIAPF2**	AI/AO	A	**Port 3 Transmit/Receive Positive/Negative** Connect directly to XFMR without any pull-down terminators, such as resistors or capacitors, required.
J41	**TPIANF2**	AI/AO	A	
G38	**TPIBPF2**	AI/AO	A	
D37	**TPIBNF2**	AI/AO	A	
B34	**TPICPF2**	AI/AO	A	
E34	**TPICNF2**	AI/AO	A	
F31	**TPIDPF2**	AI/AO	A	
J29	**TPIDNF2**	AI/AO	A	
G26	**TPIAPF3**	AI/AO	A	**Port 4 Transmit/Receive Positive/Negative** Connect directly to XFMR without any pull-down terminators, such as resistors or capacitors, required.
D26	**TPIANF3**	AI/AO	A	
E22	**TPIBPF3**	AI/AO	A	
B21	**TPIBNF3**	AI/AO	A	
F18	**TPICPF3**	AI/AO	A	
J16	**TPICNF3**	AI/AO	A	
G14	**TPIDPF3**	AI/AO	A	
D13	**TPIDNF3**	AI/AO	A	
Ethernet Port Calibration				
AF65	**RCAL**	AI/AO	A	**Calibration for all GPHY Ethernet Ports** Connect to 22 kΩ resistor to GND.

2.4 Digital Signal Ball (Pin) Reset Property and Drive Strength

This section describes the reset values and drive strength of the digital IO pins.

Table 35 defines the abbreviations used in this section.

- The **Reset State** is the logic level of a pin during the POR phase (internal Power_on Reset signal in Figure 15 is low).
- The **Reset Release State** is the logic level of a pin after the POR signal (internal Power_on Reset signal in Figure 15) transition from low to high.

Table 35 Abbreviations for the Reset Property

Abbreviation	Description
Z	High impedance
L	Pin not driven, internal pull-down
H	Pin not driven, internal pull-up

2.4.1 SDIO

The SDIO interface needs to support either 3.3 V or 1.8 V operation.

Table 36 SDIO Signals

Name	Reset State	Reset Release State	Drive Strength (mA)	Voltage Domain
GPIO74	L	Z	4, programmable (2, 4, 8, 12 mA)	3.3/1.8 V
GPIO75	L	Z	4, programmable (2, 4, 8, 12 mA)	3.3/1.8 V
GPIO76	L	Z	4, programmable (2, 4, 8, 12 mA)	3.3/1.8 V
GPIO77	L	Z	4, programmable (2, 4, 8, 12 mA)	3.3/1.8 V
GPIO78	L	Z	4, programmable (2, 4, 8, 12 mA)	3.3/1.8 V
GPIO79	L	Z	4, programmable (2, 4, 8, 12 mA)	3.3/1.8 V
GPIO80	L	Z	4, programmable (2, 4, 8, 12 mA)	3.3/1.8 V
GPIO81	L	Z	4, programmable (2, 4, 8, 12 mA)	3.3/1.8 V

2.4.2 TDM

Table 37 describes the TDM interface multiplexing.

Table 37 TDM Signals

Name	Reset State	Reset Release State	Drive Strength (mA)	Voltage Domain
C55_TDM1_DCL	L	Z	4	3.3 V
C55_TDM1_DI	L	Z	4	3.3 V
C55_TDM1_DO	L	Z	4	3.3 V
C55_TDM1_FSC	L	Z	4	3.3 V
C55_TDM0_DO	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
C55_TDM0_FSC	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V

2.4.3 16/8-bit NAND Flash

Table 38 describes the 16/8-bit NAND flash interface.

Table 38 NAND Interface Signals

Name	Reset State	Reset Release State	Drive Strength (mA)	Voltage Domain
GPIO50	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO51	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO52	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO53	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO54	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO55	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO56	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO57	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO23	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO49	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO13	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO24	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO48	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO59	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO60	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO61	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V

2.4.4 Serial Peripheral Interface (SPI)

The SPI interface is targeted for serial flash support.

Table 39 SPI/GPIO Interface Signals

Name	Reset State	Reset Release State	Drive Strength (mA)	Voltage Domain
GPIO10	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO11	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO14	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO19	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO15	L	Z[1)]	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO16	L	Z[2)]	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO18	L	Z[3)]	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO17	L	Z[4)]	2, programmable (2, 4, 8, 12 mA)	3.3 V

1) ROM programs it to SPI0_CS1_N when boot strapping selects SPI. In this case, the ROM programs the pin according to the needs of the SPI master interface.
2) ROM programs it to SPI0_RX when boot strapping selects SPI. In this case, the ROM programs the pin according to the needs of the SPI master interface.
3) ROM programs it to SPI0_CLK when boot strapping selects SPI. In this case, the ROM programs the pin according to the needs of the SPI master interface.
4) ROM programs it to SPI0_TX when boot strapping selects SPI. In this case, the ROM programs the pin according to the needs of the SPI master interface.

2.4.5 UART

It is possible to connect several peripherals to the UART. One UART interface is intended for debugging (OS console) purposes when console on UART is enabled.

Table 40 UART Interface Signals

Name	Reset State	Reset Release State	Drive Strength (mA)	Voltage Domain
UART1_RX	Z	Z	2 mA	3.3 V
UART1_TX	Z	L	2 mA	3.3 V
UART0_RX	Z	Z	2 mA	3.3 V
UART0_TX	Z	L	2 mA	3.3 V
UART2_RX	Z	Z	2 mA	3.3 V
UART2_TX	Z	L	2 mA	3.3 V
UART3_RX	Z	Z	2 mA	3.3 V
UART3_TX	Z	L	2 mA	3.3 V

2.4.6 Peripherals, General Purpose I/O, Reset

This section describes the peripherals and general purpose I/O pins.

For all other GPIO pins not mentioned in Section 2.4.1, Section 2.4.2, Section 2.4.3, Section 2.4.4, and Section 2.4.5, follow the settings given in Table 41.

Table 41 GPIO Interface Signals

Name	Reset State	Reset Release State	Drive Strength (mA)	Voltage Domain
GPIO0	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO1	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO2	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO3	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO4	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO5	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO6	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO7	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO8	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO9	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO21	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO22	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO43	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO42	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V

3 Pin Description of MxL25641

This chapter describes the pin diagram for MxL25641.

3.1 List of Interfaces

Table 42 summarizes the supported interfaces.

Table 42 External Signals

Interface	Link
PCIe 4.0/XFI Lane 0/1/2/3	Table 46 Table 47 Table 48 Table 49
USB 3.2 Port 1	Table 52
GPIO Port includes: • LED Controller • External Interrupts • System Clocks • 8-bit NAND • PCM • SPI • I2S • I2C • QSPI	Table 53
JTAG	Table 55
32-bit DDR3L/4/LPDDR4 SDRAM	Table 56
Clock	Table 57
Reset	Table 58
Power	Table 59
Fuse	Table 62

3.2 Ball Diagram for MxL25641

The color code means:

- Green = DDR Interface
- Red = Core Power
- Light Red = Analog and IO Power
- Orange = PCIe/XFI
- Blue = XFI/SFI/SGMII Signals
- Pink = Clock, Reset, Dying Gasp
- Yellow = GPIO, eMMC, EJTAG Interface
- Grey = Ground, Fuse, Reserved
- White = USB

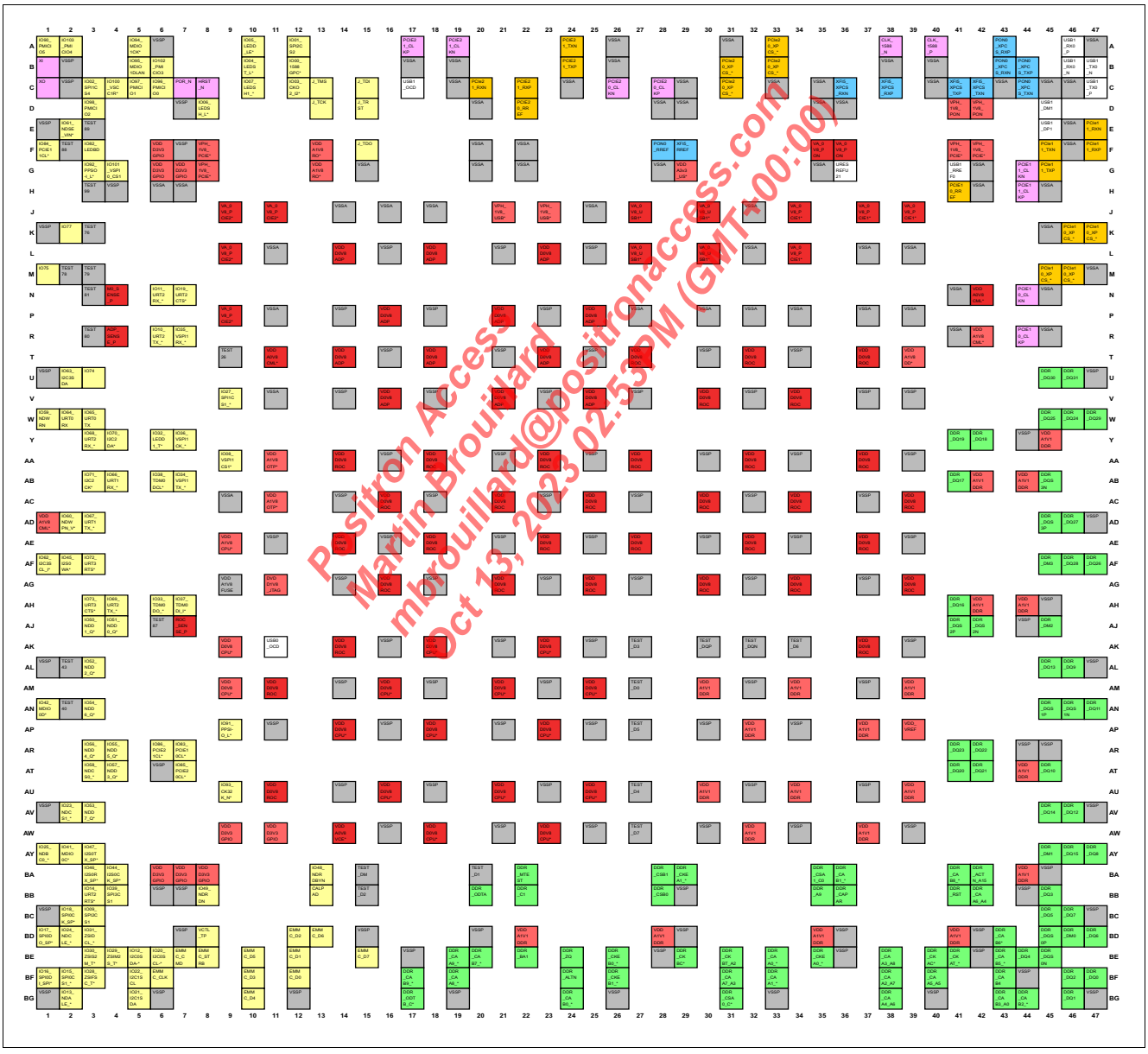


Figure 6 Ball Diagram for PG-FCBGA-577 (Top View)

3.3 Ball (Pin) Function Description

Use these abbreviations for the I/O table.

Table 43 Abbreviations for Pin Type

Abbreviations	Description
I	Input only, digital levels
O	Output only, digital levels
I/O	Bidirectional input/output signal, digital levels
AI	Input only, analog levels
AO	Output only, analog levels
AI/O	Bidirectional, analog levels
PWR	Power
GND	Ground
MCL	Must be connected to Low (JEDEC Standard).
MCH	Must be connected to High (JEDEC Standard).
NU	Not Usable (JEDEC Standard)
NC	Not Connected (JEDEC Standard). Do not connect in board design.
Prg	Programmable (either input or output)

Table 44 Abbreviations for Buffer Type

Abbreviations	Description
A	Analog
Z	High impedance
Prg	Programmable (OD/PP, PU/PD are programmable)
PU1	Pull up (internal, weak)
PD1	Pull down (internal, weak)
PD2	Pull down (internal, weaker)
TS	Tristate capability: The corresponding pin has 3 operational states: Low, high and high-impedance.
OD	Open Drain. The corresponding pin has 2 operational states, active low and tristate, and allows multiple devices to share as a wire-OR. An external pull-up is required to sustain the inactive state until another agent drives it, and must be provided by the central resource.
OC	Open Collector
PP	Push-Pull. The corresponding pin has 2 operational states: Active-low and active-high (identical to output with no type attribute).
OD/PP	Open-Drain or Push-Pull. The corresponding pin can be configured either as an output with the OD attribute or as an output with the PP attribute.
ST	Schmitt-Trigger characteristics
TTL	TTL characteristics

Table 45 GPIO Interface Multiplexing

The individual pin function after reset is flagged by a note ('Note: Default after Reset.'), otherwise it is the first column field, in other words GPIO function.

PORTMUX bit [00] 0	PORTMUX bit [01] 1	PORTMUX bit [10] 2	PORTMUX bit [11] 3
GPIO00	GPC1_1588	–	SPI1_CS2_N
GPIO01	–	–	SPI2_CS2_N
GPIO02	–	–	SPI1_CS4_N
GPIO03	CLKOUT2	–	I2S_CLK1
GPIO4	LED_ST	LED_DD0	IO04_LEDST_LEDDD0_SPISLVCK *Note: Default after Reset.*
GPIO5	LED_D	LED_DD1	IO05_LEDD_LEDDD1_SPISLVI *Note: Default after Reset.*
GPIO6	LED_SH	LED_DD2	IO06_LEDSH_LEDDD2_SPISLVO *Note: Default after Reset.*
GPIO07	LED_SH1	CLKOUT1	FAN_CTRL_I
GPIO08	C55_SPI1_CS1	CLKOUT0	-
GPIO9	SPI2_CS1_N	PON_Tx_SD	SPI2_CS1_N *Note: Default after Reset.*
GPIO10	SPI1_TX	SSI1_TX	UART2_TX
GPIO11	SPI1_RX	SSI1_RX	UART2_RX
GPIO12	–	I2C0SDA-PMIC *Note: Default after Reset.*	SPI0_CS6_N
GPIO13	NDALE *Note: Default after Reset.*	--	SPI1_CS3_N
GPIO14	SPI1_CS0_N	SLIC_RST1	UART2_RTS
GPIO15	SPI0_CS1_N	–	C55_SPI1_CS0
GPIO16	SPI0_DIN	–	C55_SPI1_RX
GPIO17	SPI0_DOUT	–	C55_SPI1_TX
GPIO18	SPI0_CLK	–	C55_SPI1_CLK
GPIO19	SPI1_CLK	SSI1_CLK	UART2_CTS
GPIO20	–	I2C0SCL-PMIC *Note: Default after Reset.*	SPI1_CS5_N
GPIO21	–	I2C_SDA1	–
GPIO22	–	I2C_SCL1	–
GPIO23	LED_ST1 *Note: Default after Reset.*	LED_ST1	–

Table 45 GPIO Interface Multiplexing (cont'd)

The individual pin function after reset is flagged by a note ('Note: Default after Reset.'), otherwise it is the first column field, in other words GPIO function.

PORTMUX bit [00] 0	PORTMUX bit [01] 1	PORTMUX bit [10] 2	PORTMUX bit [11] 3
GPIO24	NDCLE *Note: Default after Reset.*	–	SPI0_CS4_N
GPIO25	NDBC0 *Note: Default after Reset.*	–	GPC2_1588
GPIO27	SLIC_RST0	SPI1_CS1_N	C55_RESET0
GPIO28	-	TDM1_FSC	–
GPIO29	-	TDM1_DO	–
GPIO30	-	TDM1_DI	–
GPIO31	-	TDM1_DCL	–
GPIO32	LED_D1	I2S_WA1	TDM0_FSC
GPIO33	–	TDM0_DO	I2S_TX1
GPIO34	SSI0_TX	C55_SPI1_TX	-
GPIO35	SSI0_RX	C55_SPI1_RX	-
GPIO36	SSI0_CLK	C55_SPI1_CLK	-
GPIO37	–	TDM0_DI	I2S_RX1
GPIO38	–	TDM0_DCL	I2S_CLK1
GPIO39	PON_LOS	SPI3_CS1_N	–
GPIO41	MDIO0_WANC	FAN_CTRL_IN	LED_DD4
GPIO42	MDIO0_WAN	LED_DD5	FAN_CTRL_OUT
GPIO44	I2S_CLK0	SPI2_CLK	–
GPIO45	I2S_WA0	SPI2_TX	–
GPIO46	I2S_RX0	SPI2_RX	–
GPIO47	I2S_TX0	SPI2_CS0_N	–
GPIO48	NDRDBYN *Note: Default after Reset.*	–	–
GPIO49	NDRDN *Note: Default after Reset.*	–	–
GPIO50	NDD1 *Note: Default after Reset.*	–	QSPI_D1
GPIO51	NDD0 *Note: Default after Reset.*	–	QSPI_D0
GPIO52	NDD2 *Note: Default after Reset.*	–	QSPI_D2
GPIO53	NDD7 *Note: Default after Reset.*	–	QSPI_D3

Table 45 GPIO Interface Multiplexing (cont'd)

The individual pin function after reset is flagged by a note ('Note: Default after Reset.'), otherwise it is the first column field, in other words GPIO function.

PORTMUX bit [00] 0	PORTMUX bit [01] 1	PORTMUX bit [10] 2	PORTMUX bit [11] 3
GPIO54	NDD6 *Note: Default after Reset.*	–	QSPI_CLK
GPIO55	NDD5 *Note: Default after Reset.*	–	QSPI_RST0
GPIO56	NDD4 *Note: Default after Reset.*	–	QSPI_CS0
GPIO57	NDD3 *Note: Default after Reset.*	–	QSPI_CS1
GPIO58	NDCS0 *Note: Default after Reset.*	–	QSPI_RST1
GPIO59	NDWRN *Note: Default after Reset.*	–	–
GPIO60	NDWPN *Note: Default after Reset.*	SLIC2C55_INT1	I2S_TX1
GPIO61	NDSE *Note: Default after Reset.*	SLIC2C55_INT0	I2S_WA1
GPIO62	I2C_3_SCL	–	I2S_RX1
GPIO63	I2C_3_SDA	–	–
GPIO64	IO64_URT0RX *Note: Default after Reset.*	–	–
GPIO65	IO65_URT0TX *Note: Default after Reset.*	–	–
GPIO66	UART1_RX	C55_TDM2_FSC	TDM2_FSC
GPIO67	UART1_TX	C55_TDM2_DO	TDM2_DO
GPIO68	UART2_RX	C55_TDM2_DI	TDM2_DI
GPIO69	UART2_TX	C55_TDM2_DCL	TDM2_DCL
GPIO70	I2C_SDA2	UART3_RX *Note: Default after Reset.*	SPI3_CS0_N
GPIO71	I2C_SCL2	UART3_TX *Note: Default after Reset.*	SPI3_TX
GPIO72	PON_TX_Fault	UART3_RTS *Note: Default after Reset.*	SPI3_RX
GPIO73	PON_TX_DISABLE	UART3_CTS *Note: Default after Reset.*	SPI3_CLK
GPIO74	--	--	--
GPIO75	--	--	--

Table 45 GPIO Interface Multiplexing (cont'd)

The individual pin function after reset is flagged by a note ('Note: Default after Reset.'), otherwise it is the first column field, in other words GPIO function.

PORTMUX bit [00] 0	PORTMUX bit [01] 1	PORTMUX bit [10] 2	PORTMUX bit [11] 3
GPIO77	--	--	--
GPIO82	–	--	LED_BD
GPIO83	–	PCI10_CLK_REQ[1)]	–
GPIO84	–	PCI11_CLK_REQ[1)]	–
GPIO85	–	PCI20_CLK_REQ[1)]	–
GPIO86	–	PCI21_CLK_REQ[1)]	–
GPIO90	PMIC_GPIO5	--	–
GPIO91	PPS1_IN_OUT	LED_DD6	–
GPIO92	PPS2_IN_OUT	LED_DD7	–
GPIO93	CLK32K	NTR	LED_DD9
GPIO94	MDIO1_CLK	–	–
GPIO95	MDIO1_D	–	–
GPIO96	PMICIO0 *Note: Default after Reset.*	–	–
GPIO97	PMICIO1 *Note: Default after Reset.*	–	–
GPIO98	PMICIO2 *Note: Default after Reset.*	–	–
GPIO100	C55_SCC1_RESET1	–	–
GPIO101	C55_SPI0_CS1	–	--
GPIO102	PMICIO3 *Note: Default after Reset.*	–	–
GPIO103	PMICIO4 *Note: Default after Reset.*	–	–

1) It is possible to use this pin to give HW indication of PCI clock request to this particular lane when programmed in this alternative function mode. Otherwise, this pin is a general GPIO.

3.3.1 PCIe 4.0/XFI Interface #10 of HSIO1

Table 46 describes the PCI Express interface. These pins are multiplexed with 10G XFI interface pins. For details refer to PCIe 4.0/SATA 3.2/XFI Combo Subsystem.

Table 46 PCIe 4.0/XFI Interface #10 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
M45	**PCIe10_XPCS_TXP**	AO	–	**Transmit Data Pair (Differential)**
M46	**PCIe10_XPCS_TXN**	AO	–	
K46	**PCIe10_XPCS_RXP**	AI	–	**Receive Data Pair (Differential)**
K47	**PCIe10_XPCS_RXN**	AI	–	
H41	**PCIE10_RREF**	AI/AO	–	**Reference Resistor for HSIO module 1** *Note: 1% accuracy required.*

3.3.2 PCIe 4.0/XFI Interface #11 of HSIO1

Table 47 describes the PCI Express interface. For details refer to PCIe 4.0/SATA 3.2/XFI Combo Subsystem.

Table 47 PCIe 4.0 Interface #11 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
G45	**PCIe11_XPCS_TXP**	AO	–	**Transmit Data Pair (Differential)**
F45	**PCIe11_XPCS_TXN**	AO	–	
F47	**PCIe11_XPCS_RXP**	AI	–	**Receive Data Pair (Differential)**
E47	**PCIe11_XPCS_RXN**	AI	–	

3.3.3 PCIe 4.0/XFI Interface #20 of HSIO2

Table 48 describes the PCI Express interface. These pins are multiplexed with 10G XFI interface pins. For details refer to PCIe 4.0/SATA 3.2/XFI Combo Subsystem.

Table 48 PCIe 4.0/XFI Interface #20 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
B33	**PCIe20_XPCS_TXP**	AO	–	**Transmit Data Pair (Differential)**
A33	**PCIe20_XPCS_TXN**	AO	–	
B31	**PCIe20_XPCS_RXP**	AI	–	**Receive Data Pair (Differential)**
C31	**PCIe20_XPCS_RXN**	AI	–	
D22	**PCIE20_RREF**	AI/AO	–	**Reference Resistor for HSIO module 2** *Note: 1% accuracy required.*

3.3.4 PCIe 4.0/XFI Interface #21 of HSIO2

Table 49 describes the PCI Express interface. For details refer to PCIe 4.0/SATA 3.2/XFI Combo Subsystem.

Table 49 PCIe 4.0 Interface #21 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
B24	**PCIE21_XPCS_TXP**	AO	–	**Transmit Data Pair (Differential)**
A24	**PCIE21_XPCS_TXN**	AO	–	
C22	**PCIE21_XPCS_RXP**	AI	–	**Receive Data Pair (Differential)**
C20	**PCIe21_XPCS_RXN**	AI	–	

3.3.5 PON_XFI Interface #4

Table 50 describes the PON_XFI interface.

Table 50 PON_XFI#4 Interface Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
B44	**PON0_XPCS_TXP**	AO	–	**Transmit Data Pair (Differential)**
C44	**PON0_XPCS_TXN**	AO	–	
A43	**PON0_XPCS_RXP**	AI	–	**Receive Data Pair (Differential)**
B43	**PON0_XPCS_RXN**	AI	–	
F28	**PON0_RREF**	AI/AO	–	**Reference Resistor** *Note: 1% accuracy required.*

3.3.6 XFI Interface #5

Table 51 describes the XFI interface.

Table 51 XFI#5 Interface Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Signal Group Analog				
C41	**XFI5_XPCS_TXP**	AO	–	**Transmit Data Pair (Differential)**
C42	**XFI5_XPCS_TXN**	AO	–	
C38	**XFI5_XPCS_RXP**	AI	–	**Receive Data Pair (Differential)**
C36	**XFI5_XPCS_RXN**	AI	–	
F29	**XFI5_RREF**	AI/AO	–	**Reference Resistor** *Note: 1% accuracy required.*

3.3.7 USB Port 1

Table 52 describes the USB interface.

Table 52 USB1 Signals

Ball No.	Name	Pin Type	Buffer Type	Function
D45	**USB1_DM1**	AI/AO	–	**D- for USB 2.0**
E45	**USB1_DP1**	AI/AO	–	**D+ for USB 2.0**
C47	**USB1_TX0_P**	AO	–	**Transmit Data Pair (Differential) for USB 3.2 Gen 2 x1**
B47	**USB1_TX0_N**	AO	–	
A46	**USB1_RX0_P**	AI	–	**Receive Data Pair (Differential) for USB 3.2 Gen 2 x1**
B46	**USB1_RX0_N**	AI	–	
G41	**USB1_RREF0**	AI/AO	–	**Reference Resistor** *Note: 1% accuracy required. Applicable to USB 3.2 PHY Interface.*
G36	**URESREFU21**	AI/AO	–	**Reference Resistor** *Note: 1% accuracy required. Applicable to USB 2.0 PHY Interface.*
C17	**USB1_OCD**	AI	–	**USB Overcurrent Sensing Circuit** The overcurrent detection measures the drop on the supply voltage. When the connected USB device draws too much power, the sensing circuit generates an interrupt and a signal. The CPU can switch off the power supply to the USB device in such an interrupt event. The signal goes to dedicated GPIO which can be used to switch off USB supply.
AK11	**USB0_OCD**	AI	–	**USB Overcurrent Sensing Circuit / Dying Gasp** The overcurrent detection measures the drop on the supply voltage. When the connected USB device draws too much power, the sensing circuit generates an interrupt and a signal. The CPU can switch off the power supply to the USB device in such an interrupt event. The signal goes to dedicated GPIO which can be used to switch off USB supply. Another alternative use for the OCD module is to create a dying gasp circuit as described in **Overcurrent Detection Comparator**.

3.3.8 GPIO/I2C/I2S/SPI/PCM

Table 53 describes the GPIO interface.

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals

Ball No.	Name	Pin Type	Buffer Type	Function
B12	**IO00_1588GPC1_SPI1CS2**	Prg	Prg	**Default after Reset: GPIO00**
	GPIO00	Prg		**General Purpose I/O Function[PMUX0]** This function is configurable via GPIO register.
	GPC1_1588	I/O		**IEEE1588 Clock Input/Output**
	SPI1_CS2_N	O		**SPI1 Chip Select 2**
A12	**IO01_SPI2CS2**	Prg	Prg	**Default after Reset: GPIO01**
	GPIO01	Prg		**General Purpose I/O Function[PMUX1]** This function is configurable via GPIO register.
	SPI2_CS2_N	O		**SPI2 Chip Select 2**
C3	**IO02_SPI1CS4**	Prg	Prg	**Default after Reset: GPIO02**
	GPIO02	Prg		**General Purpose I/O Pin[PMUX2]** This function is configurable via GPIO register.
	OCD_OUT0	O		**Overcurrent Detection 0 Output** The OCD comparator output connected to CPU interrupt is as well connected to this alternate function. In USB application it can be used to shut down the supply on the USB Host port as shown in **USB Host and Device Mode System Design Example**.
	SPI1_CS4_N	O		**SPI1 Chip Select 4**
C12	**IO03_CKO2_I2S1CLK**	Prg	Prg	**Default after Reset: GPIO03**
	GPIO03			**General Purpose I/O Function[PMUX3]** This function is configurable via GPIO register. *Note: This pin reads in pin strapping information during reset for Boot 1 Boot Mode Strap IO (BOOT1 of BOOT[3..0])*
	CLKOUT2	O		**Clock Output 25 MHz** The pin can be configured to the clock with frequencies of 25 MHz.
	I2S_CLK1	I/O		**I2S Interface 1 Clock**

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
B10	**IO04_LEDST_LEDDD0_SPISLVCK**	I	Prg	**Default after Reset: SPI Slave clock** *Note: This function is not available in the productive device.*
	LED_ST	O		**LED Strobe** This is the store signal for the external shift register. It can be a clock or a pulse. Configuration is done in the LED controller register LED_CONx.
	LED_DD0	O		**Hardware Indication LED Pin 0**
	GPIO4	Prg		**General Purpose I/O Function[PMUX4]** This function is configurable via GPIO register.
A10	**IO05_LEDD_LEDDD1_SPISLVI**	I	Prg	**Default after Reset: SPI Slave RX** *Note: This function is not available in the productive device.*
	LED_D	O		**LED Data** This is the serial data output for the external shift register.
	LED_DD1	O		**Hardware Indication LED Pin 1**
	GPIO5	Prg		**General Purpose I/O Function[PMUX5]** This function is configurable via GPIO register.
D8	**IO06_LEDSH_LEDDD2_SPISLVO**	O	Prg	**Default after Reset: SPI Slave TX** *Note: This function is not available in the productive device.*
	LED_SH	O		**LED Shift Clock** This is the shift clock for the external shift register.
	LED_DD2	O		**Hardware Indication LED Pin 2**
	GPIO6	Prg		**General Purpose I/O Function[PMUX6]** This function is configurable via GPIO register.
C10	**IO07_LEDSH1_CKOUT1_FANIN**	Prg	Prg	**Default after Reset: GPIO07**
	GPIO07	Prg		**General Purpose I/O Function[PMUX7]** This function is configurable via GPIO register.
	LED_SH1	O		**LED Shift Clock 1** This is the shift clock for the external shift register.
	CLKOUT1	O		**Clock Output 40 MHz**
	FAN_CTRL_I	I		**Fan Control Input**

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
AA9	**IO08_VSPI1CS1_ZSIFSC_CKO0**	Prg	Prg	**Default after Reset: GPIO08**
	GPIO08	Prg		**General Purpose I/O Function[PMUX8]** This function is configurable via GPIO register.
	C55_SPI1_CS1	O		**C55 SPI1 Chip Select 1**
	CLKOUT0	O		**Clock Output 8.192 MHz** The pin can be configured to the frequency of 8.192 MHz.
BC3	**IO09_SPI2CS1**	I	Prg	**Default after Reset: SPI Slave CS** *Note: This function is not available in the productive device.*
	PON_Tx_SD	I		**PON TX SD**
	SPI2_CS1_N	O		**SPI2 Chip Select 1**
	GPIO9	Prg		**General Purpose I/O Function[PMUX9]** This function is configurable via GPIO register.
R6	**IO10_URT2TX_SPI1TX_SSI1TX**	Prg	Prg	**Default after Reset: GPIO10**
	GPIO10			**General Purpose I/O Function[PMUX10]** This function is configurable via GPIO register.
	SPI1_TX	O		**SPI1 Transmit** This pin function is only available in SPI mode.
	SSI1_TX	O		**SmartSLIC1 Transmit**
	UART2_TX	O		**UART2 Transmit**
N6	**IO11_URT2RX_SPI1RX_SSI1RX**	Prg	Prg	**Default after Reset: GPIO11**
	GPIO11	Prg		**General Purpose I/O Function[PMUX11]** This function is configurable via GPIO register.
	SPI1_RX	I		**SPI 1 Receive**
	SSI1_RX	I		**SmartSLIC1 Receive**
	UART2_RX	I		**UART2 Receive**
BE5	**IO12_I2C0SDA-PMIC**	I/O	Prg	**Default after Reset: I2C0 SDA**
	GPIO12	Prg		**General Purpose I/O [PMUX12]** This function is configurable via GPIO register.
	SPI0_CS6_N	O		**SPI 0 Chip Select 6**
	I2C0SDA-PMIC	I/O		**I2C0 SDA** Serial Data IN/OUT

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
BG2	**IO13_NDALE_SPI1CS3**	O	Prg	**Default after Reset: NAND ALE**
	GPIO13	Prg		**General Purpose I/O Function[PMUX13]** This function is configurable via GPIO register.
	SPI1_CS3_N	O		**SPI 1 Chip Select 3**
	NDALE	O		**NAND Flash Address Latch Enable** The Address Latch enable loads an address into the target.
BB3	**IO14_URT2RTS_SPI1CS0_VRST1**	Prg	Prg	**Default after Reset: GPIO14**
	GPIO14	Prg		**General Purpose I/O Function[PMUX14]** This function is configurable via GPIO register. *Note: This pin reads in pin strapping information during reset for Boot 3 Boot Mode Strap IO (BOOT3 of BOOT[3..0])*
	SPI1_CS0_N	O		**SPI1 Chip Select 0** This pin function is available in SPI 1.
	SLIC_RST1	O		**SmartSLIC1 Interface Reset** This pin function is only available in SSI mode.
	UART2_RTS	O		**UART2 HW Flow Control Signal**
BF2	**IO15_SPI0CS1_SPI1CS0**	Prg	Prg	**Default after Reset: GPIO15**
	GPIO15	Prg		**General Purpose I/O Function[PMUX15]** This function is configurable via GPIO register.
	SPI0_CS1_N	I/O		**SPI0 Chip Select 1** This pin output function is only available in SPI master mode. When it is slave, it is an input. This CS is used for Serial Flash boot.
	C55_SPI1_CS0	O		**C55 Voice SPI1 Chip Select 0**
BF1	**IO16_SPI0DI_SPI1RX**	Prg	Prg	**Default after Reset: GPIO16**
	GPIO16	Prg		**General Purpose I/O Function[PMUX16]** This function is configurable via GPIO register.
	SPI0_DIN	I/O		**SPI0 Data Input** In Slave mode, this pin is output. In Master mode, this pin is input.
	C55_SPI1_RX	I		**C55 Voice SPI1 RX**
BD1	**IO17_SPI0DO_SPI1TX**	Prg	Prg	**Default after Reset: GPIO17**
	GPIO17			**General Purpose I/O Function[PMUX17]** This function is configurable via GPIO register.
	SPI0_DOUT	I/O		**SPI0 Data Output** In Slave mode, this pin is input. In Master mode, this pin is output.
	C55_SPI1_TX	O		**C55 Voice SPI1 TX**

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
BC2	**IO18_SPI0CK_SPI1CK**	Prg	Prg	**Default after Reset: GPIO18**
	GPIO18	Prg		**General Purpose I/O Function[PMUX18]** This function is configurable via GPIO register.
	SPI0_CLK	I/O		**SPI0 Clock** In Slave mode, the clock is input. In Master mode, the clock is output and only active when data is transmitted.
	C55_SPI1_CLK	O		**C55 Voice SPI1 Clock**
N7	**IO19_URT2CTS_SPI1CK_SSI1CK**	Prg	Prg	**Default after Reset: GPIO19**
	GPIO19	Prg		**General Purpose I/O Function[PMUX19]** This function is configurable via GPIO register. *Note: This pin reads in pin strapping information during reset for Boot 2 Boot Mode Strap IO (BOOT2 of BOOT[3..0])*
	SPI1_CLK	O		**SPI1 Clock** This pin function is available in SPI 1.
	SSI1_CLK	I		**SmartSLIC1 Interface Clock** This pin function is only available in SSI mode.
	UART2_CTS	O		**UART2 HW flow Control Signal**
BE6	**IO20_I2C0SCL-PMIC**	O	Prg	**Default after Reset: I2C0 SCL**
	GPIO20	Prg		**General Purpose I/O[PMUX20]**
	SPI1_CS5_N	O		**SPI #1 Chip Select 5**
	I2C0SCL-PMIC	O		**I2C0 Serial Clock PMIC**
BG5	**IO21_I2C1SDA**	Prg	Prg	**Default after Reset: GPIO21**
	GPIO21	Prg		**General Purpose I/O Function[PMUX21]** This function is configurable via GPIO register.
	I2C_SDA1	I/O		**I2C1 Serial Data (SDA)**
BF5	**IO22_I2C1SCL**	Prg	Prg	**Default after Reset: GPIO22**
	GPIO22	Prg		**General Purpose I/O Function[PMUX22]** This function is configurable via GPIO register.
	I2C_SCL1	O		**I2C1 Serial Clock Line (SCL)**
AV2	**IO23_NDCS1_LEDST1**	O	Prg	**Default after Reset: NDCS1**
	GPIO23	Prg		**General Purpose I/O Function[PMUX23]** This function is configurable via GPIO register.
	LED_ST1	O		**LED SSO 1 Strobe**
	NDCS1	O		**NAND Flash Chip Select 1** This NAND chip select is used by ROM for booting.

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
BD2	**IO24_NDCLE_SPI0CS4**	O	Prg	**Default after Reset: NDCLE**
	GPIO24	Prg		**General Purpose I/O Function[PMUX24]** This function is configurable via GPIO register.
	SPI0_CS4_N	O		**SPI 0 Chip Select 4**
	NDCLE	O		**NAND Flash Command Latch Enable**
AY1	**IO25_NDBC0_1588GPC2**	O	Prg	**Default after Reset: NAND BC0**
	NDBC0			**NAND BC0**
	GPIO25	I/O		**General Purpose I/O Function[PMUX25]** This function is configurable via GPIO register.
	GPC2_1588	I/O		**General Purpose Clock 2 for 1588**
V9	**IO27_SPI1CS1_VRST0**	I/O	Prg	**Default after Reset: GPIO27**
	GPIO27	Prg		**General Purpose I/O Function[PMUX27]** This function is configurable via GPIO register.
	SLIC_RST0	O		**SLIC Reset**
	SPI1_CS1_N	O		**SPI #1 Chip Select 1**
	C55_RESET0	O		**C55 SPI Reset 0**
BF3	**IO28_ZSIFSC_TDM1FSC**	O	Prg	**Default after Reset: GPIO28**
	GPIO28			**General Purpose I/O[PMUX28]** Supplied by 3.3 V only.
	TDM1_FSC	I/O		**TDM Frame Sync**
BE4	**IO29_ZSIM2S_TDM1DO**	O	Prg	**Default after Reset: GPIO29**
	GPIO29			**General Purpose I/O[PMUX29]** Supplied by 3.3 V only.
	TDM1_DO	O		**TDM Data Output**
BE3	**IO30_ZSIS2M_TDM1DI**	O	Prg	**Default after Reset: GPIO30**
	GPIO30	Prg		**General Purpose I/O[PMUX30]** Supplied by 3.3 V only.
	TDM1_DI	I		**TDM Data Input**
BD3	**IO31_ZSIDCL_TDM1DCL**	O	Prg	**Default after Reset: GPIO31**
	GPIO31	Prg		**General Purpose I/O[PMUX31]** Supplied by 3.3 V only.
	TDM1_DCL	I/O		**TDM Data Clock**

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
Y6	**IO32_LEDD1_TDM0FSC_I2S1WA**	I/O	Prg	**Default after Reset: GPIO32**
	GPIO32	Prg		**General Purpose I/O Function[PMUX32]** This function is configurable via GPIO register.
	LED_D1			**LED Controller 1**
	I2S_WA1	O		**I2S1 WA**
	TDM0_FSC	O		**TDM Frame Sync**
AH6	**IO33_TDM0DO_I2S1TX**	I/O	Prg	**Default after Reset: GPIO33**
	GPIO33	Prg		**General Purpose I/O Function[PMUX33]** This function is configurable via GPIO register.
	TDM0_DO	O		**TDM Data output**
	I2S_TX1	O		**I2S1 Transmit**
AB7	**IO34_VSPI1TX_ZSIM2S_SSI0TX**	I/O	Prg	**Default after Reset: GPIO34**
	GPIO34	Prg		**General Purpose I/O Function[PMUX34]** This function is configurable via GPIO register.
	SSI0_TX	O		**SmartSLIC0 Interface TX**
	C55_SPI1_TX	O		**SPI1 C55 Transmit**
R7	**IO35_VSPI1RX_ZSIS2M_SSI0RX**	I/O	Prg	**Default after Reset: GPIO35**
	GPIO35	Prg		**General Purpose I/O Function[PMUX35]** This function is configurable via GPIO register.
	SSI0_RX	I		**SmartSLIC0 Interface Rreceive**
	C55_SPI1_RX	I		**SPI1 C55 Trasmit**
Y7	**IO36_VSPI1CK_ZSIDCL_SSI0CK**	Prg	Prg	**Default after Reset: GPIO36**
	GPIO36	Prg		**General Purpose I/O Function[PMUX36]** This function is configurable via GPIO register.
	SSI0_CLK	I		**SmartSLIC0 Interface Clock** This pin function is only available in SSI mode.
	C55_SPI1_CLK	O		**SPI1 C55 Clock**
AH7	**IO37_TDM0DI_I2S1RX**	I/O	Prg	**Default after Reset: GPIO37**
	GPIO37	Prg		**General Purpose I/O Function[PMUX37]** This function is configurable via GPIO register.
	TDM0_DI	I		**TDM0 Data Input**
	I2S_RX1	I		**I2S1 Receive**

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
AB6	**IO38_TDM0DCL_I2S1CK**	I/O	Prg	**Default after Reset: GPIO38**
	GPIO38	Prg		**General Purpose I/O Function[PMUX38]** This function is configurable via GPIO register.
	TDM0_DCL	O		**DCL of TDM0**
	I2S_CLK1	O		**I2S Clock 1**
BB4	**IO39_SPI3CS1**	I/O	Prg	**Default after Reset: GPIO39**
	GPIO39	Prg		**General Purpose I/O Function[PMUX39]** This function is configurable via GPIO register.
	PON_LOS	O		**PON_Rx_LOS**
	SPI3_CS1_N	O		**SPI3 Chip Select 1**
AY2	**IO41_MDIO0CKWAN_FANIN**	I/O	Prg	**Default after Reset: GPIO41**
	GPIO41	Prg		**General Purpose I/O Function[PMUX41]** This function is configurable via GPIO register.
	MDIO0_WANC	O		**WAN MDIO Clock**
	FAN_CTRL_IN	I		**Fan Control Input**
	LED_DD4			**Direct LED DATA 4**
AN1	**IO42_MDIO0DWAN_FANOUT**	I/O	Prg	**Default after Reset: GPIO42**
	GPIO42	Prg		**General Purpose I/O Function[PMUX42]** This function is configurable via GPIO register.
	MDIO0_WAN	I/O		**WAN MDIO Data**
	LED_DD5			**Direct LED DATA 5**
	FAN_CTRL_OUT	O		**FAN Controller**
BA4	**IO44_I2S0CK_SPI2CK**	I/O	Prg	**Default after Reset: GPIO44**
	GPIO44	Prg		**General Purpose I/O Function[PMUX44]** This function is configurable via GPIO register.
	I2S_CLK0	I/O		**I2S Clock 0**
	SPI2_CLK	O		**SPI2 Clock**
AF2	**IO45_I2S0WA_SPI2TX**	I/O	Prg	**Default after Reset: GPIO45**
	GPIO45	Prg		**General Purpose I/O Function[PMUX45]** This function is configurable via GPIO register.
	I2S_WA0	I/O		**I2S WA0**
	SPI2_TX	O		**SPI2 Transmit**
BA3	**IO46_I2S0RX_SPI2RX**	I/O	Prg	**Default after Reset: GPIO46**
	GPIO46	Prg		**General Purpose I/O Function[PMUX46]** This function is configurable via GPIO register.
	I2S_RX0	I/O		**I2S0 Receive**
	SPI2_RX	I		**SPI2 Receive**

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
AY3	**IO47_I2S0TX_SPI2CS0**	I/O	Prg	**Default after Reset: GPIO47**
	GPIO47	Prg		**General Purpose I/O Function[PMUX47]** This function is configurable via GPIO register.
	I2S_TX0	I/O		**I2S0 Transmit**
	SPI2_CS0_N	O		**SPI2 Chip Select 0**
BA13	**IO48_NDRDBYN**	I	Prg	**Default after Reset: NDRDBYN**
	NDRDBYN	I		**NAND Flash Ready /Busy** The Ready/Busy signal indicates the NAND device status. When low, the signal indicates that no more NAND operations are in progress.
	GPIO48	Prg		**General Purpose I/O Function[PMUX48]** This function is configurable via GPIO register.
BB8	**IO49_NDRDN**	O	Prg	**Default after Reset: NDRDN**
	NDRDN	O		**NAND Flash Read Enable** The NAND Read enable controls the latching of input data.
	GPIO49	Prg		**General Purpose I/O Function[PMUX49]** This function is configurable via GPIO register.
AJ3	**IO50_NDD1_QSPID1**	I/O	Prg	**Default after Reset: NDD1**
	NDD1	I/O		**NAND Flash Command/Address/Data Pin 1**
	GPIO50	Prg		**General Purpose I/O Function[PMUX50]** This function is configurable via GPIO register.
	QSPI_D1	I/O		**Quad SPI Data 1**
AJ4	**IO51_NDD0_QSPID0**	I/O	Prg	**Default after Reset: NDD0**
	NDD0	I/O		**NAND Flash Command/Address/Data Pin 0**
	GPIO51	Prg		**General Purpose I/O Function[PMUX51]** This function is configurable via GPIO register.
	QSPI_D0	I/O		**Quad SPI Data 0**
AL3	**IO52_NDD2_QSPID2**	I/O	Prg	**Default after Reset: NDD2**
	NDD2	I/O		**NAND Flash Command/Address/Data Pin 2**
	GPIO52	Prg		**General Purpose I/O Function[PMUX52]** This function is configurable via GPIO register.
	QSPI_D2	I/O		**Quad SPI Data 2**
AV3	**IO53_NDD7_QSPID3**	I/O	Prg	**Default after Reset: NDD7**
	NDD7	I/O		**NAND Flash Command/Address/Data Pin 7**
	GPIO53	Prg		**General Purpose I/O Function[PMUX53]** This function is configurable via GPIO register.
	QSPI_D3	I/O		**Quad SPI Data 3**

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
AN3	**IO54_NDD6_QSPICK**	I/O	Prg	**Default after Reset: NDD6**
	NDD6	I/O		**NAND Flash Command/Address/Data Pin 6**
	GPIO54	Prg		**General Purpose I/O Function[PMUX54]** This function is configurable via GPIO register.
	QSPI_CLK	I/O		**Quad SPI Clock** In Master mode, the clock is output and only active when data is transmitted.
AR4	**IO55_NDD5_QSPIRST0**	I/O	Prg	**Default after Reset: NDD5**
	NDD5	I/O		**NAND Flash Command/Address/Data Pin 5**
	GPIO55	Prg		**General Purpose I/O Function[PMUX55]** This function is configurable via GPIO register.
	QSPI_RST0	O		**Quad SPI Reset 0**
AR3	**IO56_NDD4_QSPICS0**	I/O	Prg	**Default after Reset: NDD4**
	NDD4	I/O		**NAND Flash Command/Address/Data Pin 4**
	GPIO56	Prg		**General Purpose I/O Function[PMUX56]** This function is configurable via GPIO register.
	QSPI_CS0	O		**Quad SPI Chip Select 0**
AT4	**IO57_NDD3_QSPICS1**	I/O	Prg	**Default after Reset: NDD3**
	NDD3	I/O		**NAND Flash Command/Address/Data Pin 3**
	GPIO57	Prg		**General Purpose I/O Function[PMUX57]** This function is configurable via GPIO register.
	QSPI_CS1	O		**Quad SPI Chip Select 1**
AT3	**IO58_NDCS0_QSPIRST1**	O	Prg	**Default after Reset: NDCS0**
	NDCS0	O		**NAND Flash Chip Select 0** This is NAND chip select 0.
	GPIO58	I/O		**General Purpose I/O Function[PMUX58]** This function is configurable via GPIO register.
	QSPI_RST1	O		**Quad SPI Reset 1**
W1	**IO59_NDWRN**	O	Prg	**Default after Reset: NDWRN**
	NDWRN	O		**NAND Flash Write Enable** The NAND Write enable control the sending of output data.
	GPIO59	Prg		**General Purpose I/O Function[PMUX59]** This function is configurable via GPIO register.

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
AD2	**IO60_NDWPN_VINT1_I2STX1**	O	Prg	**Default after Reset: NDWPN**
	NDWPN	O		**NAND Flash Write Protect** The Write protect signal disables flash array program and erase operations.
	GPIO60	Prg		**General Purpose I/O Function[PMUX60]** This function is configurable via GPIO register.
	SLIC2C55_INT1	-		**Interrupt 1 to C55 SLIC2**
	I2S_TX1	I/O		**I2S1 Transmit**
E2	**IO61_NDSE_VINT0_I2SWA1**	O	Prg	**Default after Reset: NDSE**
	NDSE	O		**NAND Flash Spare Area Enable** The spare area enable output controls the access of the spare area. When SE is high, the spare area is not accessible for reading or programming.This is used in legacy NAND devices.
	GPIO61	Prg		**General Purpose I/O Function[PMUX61]** This function is configurable via GPIO register.
	SLIC2C55_INT0	I		**Interrupt 0 to C55 SLIC2**
	I2S_WA1	I/O		**I2S WA1**
AF1	**IO62_I2C3SCL_I2SRX1**	I/O	Prg	**Default after Reset: GPIO62**
	GPIO62	Prg		**General Purpose I/O Function[PMUX62]** This function is configurable via GPIO register.
	I2C_3_SCL	O		**I2C3 Serial Clock Line**
	I2S_RX1	I		**I2S1 Receive**
U2	**IO63_I2C3SDA**	I/O	Prg	**Default after Reset: GPIO63**
	GPIO63	Prg		**General Purpose I/O Function[PMUX63]** This function is configurable via GPIO register.
	I2C_3_SDA	I/O		**I2C3 Serial Data**
W2	**IO64_URT0RX**	I	Prg	**Default after Reset: UART0RX**
	UART0RX	I		**UART0 RX**
	GPIO64	Prg		**General Purpose I/O Function[PMUX64]** This function is configurable via GPIO register.
W3	**IO65_URT0TX**	O	Prg	**Default after Reset: UART0TX**
	UART0TX			**UART0 TX**
	GPIO65	Prg		**General Purpose I/O Function[PMUX65]** This function is configurable via GPIO register.

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
AB4	**IO66_URT1RX_TDM2FSC**	Prg	Prg	**Default after Reset: GPIO66**
	GPIO66	Prg		**General Purpose I/O Function[PMUX66]** This function is configurable via GPIO register.
	UART1_RX	I		**UART1 Receive**
	C55_TDM2_FSC	O		**TDM2 C55 Frame Sync**
	TDM2_FSC	O		**TDM 2 Frame Sync Clock**
AD3	**IO67_URT1TX_TDM2DO**	Prg	Prg	**Default after Reset: GPIO66** *Note: This pin reads in pin strapping information during reset for Boot 0 Boot Mode Strap IO (BOOT0 of BOOT[3..0])*
	GPIO67	Prg		**General Purpose I/O Function[PMUX67]** This function is configurable via GPIO register.
	UART1_TX	O		**UART1 Transmit**
	C55_TDM2_DO	O		**TDM2 C55 Data output**
	TDM2_DO	O		**TDM2 Data output**
Y3	**IO68_URT2RX_TDM2DI**	Prg	Prg	**Default after Reset: GPIO68**
	GPIO68	Prg		**General Purpose I/O Function[PMUX68]** This function is configurable via GPIO register.
	UART2_RX	I		**UART2 Receive**
	C55_TDM2_DI	I		**TDM2 C55 Data in**
	TDM2_DI	I		**TDM 2 Data In**
AH4	**IO69_URT2TX_TDM2DCL**	Prg	Prg	**Default after Reset: GPIO69**
	GPIO69	Prg		**General Purpose I/O Function[PMUX69]** This function is configurable via GPIO register.
	UART2_TX	O		**UART2 Transmit**
	C55_TDM2_DCL	O		**TDM2 C55 Data Clock**
	TDM2_DCL	O		**TDM2 Data Clock**
Y4	**IO70_I2C2DA_URT3RX_SPI3CS0**	I	Prg	**Default after Reset: UART3_RX**
	UART3_RX	I		**UART3 Receive**
	I2C_SDA2	I/O		**SDA of I2C 2**
	GPIO70	Prg		**General Purpose I/O Function[PMUX70]** This function is configurable via GPIO register.
	SPI3_CS0_N	O		**SPI3 Chip Select 0**

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
AB3	**IO71_I2C2CK_URT3TX_ SPI3TX**	O	Prg	**Default after Reset: UART3 TX**
	UART3_TX	O		**UART3 Transmit**
	I2C_SCL2	O		**SCL of I2C 2**
	GPIO71	Prg		**General Purpose I/O Function[PMUX71]** This function is configurable via GPIO register.
	SPI3_TX	O		**SPI3 Transmit**
AF3	**IO72_URT3RTS_SPI3RX**	I	Prg	**Default after Reset: UART3 RTS**
	UART3_RTS			**UART3 Request to Send**
	GPIO72	Prg		**General Purpose I/O Function[PMUX72]** This function is configurable via GPIO register.
	PON_TX_Fault	I		**PON Transmit Fault**
	SPI3_RX	I		**SPI3 Data IN (Receive)**
AH3	**IO73_URT3CTS_SPI3CK**	O	Prg	**Default after Reset: UART3 CTS**
	UART3_CTS	O		**UART3 Clear to Send**
	GPIO73	Prg		**General Purpose I/O Function[PMUX73]** This function is configurable via GPIO register.
	PON_TX_DISABLE	O		**PON Transmit Disable**
	SPI3_CLK	O		**SPI3 Clock**
U3	**IO74**	O	Prg	**Default after Reset: Test**
	GPIO74	Prg		**General Purpose I/O Function[PMUX74]** This function is configurable via GPIO register.
M1	**IO75**	O	Prg	**Default after Reset: Test**
	GPIO75	Prg		**General Purpose I/O Function[PMUX75]** This function is configurable via GPIO register.
K2	**IO77**	Prg	Prg	**Default after Reset: GPIO77**
	GPIO77	Prg		**General Purpose I/O Function[PMUX77]** This function is configurable via GPIO register.
F3	**IO82_LEDBD**	Prg	Prg	**Default after Reset: GPIO82**
	GPIO82	Prg		**General Purpose I/O Function[PMUX82]** This function is configurable via GPIO register.
	LED_BD	I/O		**Brightness Detection Input and Control** Indicates the Brightness information and controls the external detector.
AR7	**IO83_PCIE10CLKREQ**	Prg	Prg	**Default after Reset: GPIO83**
	GPIO83	Prg		**General Purpose I/O Function[PMUX83]** This function is configurable via GPIO register.
	PCI10_CLK_REQ	O		**PCIe10 Clock Request**

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
F1	**IO84_PCIE11CLKREQ**	Prg	Prg	**Default after Reset: GPIO84**
	GPIO84	Prg		**General Purpose I/O Function[PMUX84]** This function is configurable via GPIO register.
	PCI11_CLK_REQ	I		**PECe11 Clock Request**
AT7	**IO85_PCIE20CLKREQ**	Prg	Prg	**Default after Reset: GPIO85**
	GPIO85	Prg		**General Purpose I/O Function[PMUX85]** This function is configurable via GPIO register.
	PCI20_CLK_REQ	I		**PCIe20 Clock Request**
AR6	**IO86_PCIE21CLKREQ**	Prg	Prg	**Default after Reset: GPIO86**
	GPIO86	Prg		**General Purpose I/O Function[PMUX86]** This function is configurable via GPIO register.
	PCI21_CLK_REQ	I		**PCIe21 Clock Request**
A1	**IO90_PMICIO5**	Prg	Prg	**Default after Reset: GPIO90**
	GPIO90	Prg		**General Purpose I/O Function[PMUX90]** This function is configurable via GPIO register.
	PMIC_GPIO5	I		**GPIO5 for ext PMIC** Input from PMIC to indicate PMIC interrupt events
AP9	**IO91_PPSI-O_LEDDD6**	Prg	Prg	**Default after Reset: GPIO91**
	GPIO91	Prg		**General Purpose I/O Function[PMUX91]** This function is configurable via GPIO register.
	PPS1_IN_OUT	I/O		**PPS 1 Input/Output**
	LED_DD6	O		**LED DD6**
G3	**IO92_PPSO-I_LEDDD7**	Prg	Prg	**Default after Reset: GPIO92**
	GPIO92	Prg		**General Purpose I/O Function[PMUX92]** This function is configurable via GPIO register.
	PPS2_IN_OUT	I/O		**PPS 2 Output/Input**
	LED_DD7	O		**LED DD7**
AU9	**IO93_CK32K_NTR_LEDD D9**	Prg	PrgPrg	**Default after Reset: GPIO93**
	GPIO93	Prg		**General Purpose I/O Function[PMUX93]** This function is configurable via GPIO register.
	CLK32K	O		**32 kHz Input**
	NTR	I		**NTR Clock Input**
	LED_DD9	I/O		**LED DD9**

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
A5	**IO94_MDIO1CKLAN**	Prg	Prg	**Default after Reset: GPIO94**
	GPIO94	Prg		**General Purpose I/O Function[PMUX94]** This function is configurable via GPIO register.
	MDIO1_CLK	O		**LAN MDIO 1 Clock** This function is configurable via GPIO register. Used as MDIO master to access Ethernet subsystem PHYs.
B5	**IO95_MDIO1DLAN**	Prg	Prg	**Default after Reset: GPIO95**
	GPIO95	Prg		**General Purpose I/O Function[PMUX95]** This function is configurable via GPIO register.
	MDIO1_D	I/O		**LAN MDIO 1 Data** This function is configurable via GPIO register. Used as MDIO master to access Ethernet subsystem PHYs.
C6	**IO96_PMICIO0**	I	Prg	**Default after Reset: PMICIO0**
	PMICIO0	I		**GPIO 0 to ext PMIC** Input from PMIC to indicate power good
	GPIO96	Prg		**General Purpose I/O Function[PMUX96]** This function is configurable via GPIO register.
C5	**IO97_PMICIO1**	O	Prg	**Default after Reset: PMICIO1**
	PMICIO1	O		**GPIO 1 to ext PMIC** Output to PMIC to indicate power events request
	GPIO97	Prg		**General Purpose I/O Function[PMUX97]** This function is configurable via GPIO register.
D3	**IO98_PMICIO2**	I	Prg	**Default after Reset: PMICIO2**
	PMICIO2	I		**GPIO 2 to ext PMIC** Input from PMIC to indicate CPU0 rail state
	GPIO98	Prg		**General Purpose I/O Function[PMUX98]** This function is configurable via GPIO register.
C4	**IO100_VSCC1RST1**	Prg	Prg	**Default after Reset: GPIO100**
	GPIO100	Prg		**General Purpose I/O Function[PMUX100]** This function is configurable via GPIO register.
	C55_SCC1_RESET1	O		**SPI1 Reset of C55**
G4	**IO101_VSPI0_CS1**	Prg	Prg	**Default after Reset: GPIO101**
	GPIO101	Prg		**General Purpose I/O Function[PMUX101]** This function is configurable via GPIO register.
	C55_SPI0_CS1	O		**SPI0 cs1 of C55**

Table 53 GPIO/Flash/I2C/I2S/SPI/PCM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
B6	**IO102_PMICIO3**	I	Prg	**Default after Reset: GPIO102**
	PMICIO3	I		**GPIO3 to ext PMIC** Input from PMIC to indicate CPU1 rail state
	GPIO102	Prg		**General Purpose I/O Function[PMUX102]** This function is configurable via GPIO register.
A2	**IO103_PMICIO4**	I	Prg	**Default after Reset: GPIO103**
	PMICIO4	I		**GPIO4 to ext PMIC** Input from PMIC to indicate ADP rail state
	GPIO103	Prg		**General Purpose I/O Function[PMUX103]** This function is configurable via GPIO register.

3.3.9 eMMC

The eMMc interface is intended as a Flash interface.

Table 54 eMMC Interface

Ball No.	Name	Pin Type	Buffer Type	Function
BF12	**EMMC_D0**	I/0	A	**emmc_D0** Data pin
BE12	**EMMC_D1**	I/0	A	**emmc_D1** Data pin
BD12	**EMMC_D2**	I/0	A	**emmc_D2** Data pin
BF10	**EMMC_D3**	I/0	A	**emmc_D3** Data pin
BG10	**EMMC_D4**	I/0	A	**emmc_D4** Data pin
BE10	**EMMC_D5**	I/0	A	**emmc_D5** Data pin
BD13	**EMMC_D6**	I/0	A	**emmc_D6** Data pin
BE15	**EMMC_D7**	I/0	A	**emmc_D7** Data pin
BE8	**EMMC_STRB**	I/0	A	**emmc_Strobe** Control pin
BE7	**EMMC_CMD**	I/0	A	**emmc_cmd** Control pin
BB13	**CALPAD**	I/0	A	**emmc_calibration** 10k resistor to GND
BF6	**EMMC_CLK**	I/0	A	**emmc_clk**
BD8	**VCTL_TP**	I/0	A	**emmc test point**

3.3.10 JTAG

Table 55 describes the JTAG interface.

Table 55 JTAG Signals

Ball No.	Name	Pin Type	Buffer Type	Function
D15	**J_TRST**	I	PU	**JTAG Reset** *Note:* *3. Active low.* *4. When the JTAG interface is not in use, this pin must be connected to V_{SS}.* *5. A low-to-high transition of the TRST pin latches the JTAG mode (depending on the boot strap setting of pin J_TDO).*
	J_TRST	I	PU	**JTAG Reset**
C15	**J_TDI**	I	PU	**Test Data Input**
	J_TDI	I		**JTAG Test Data Input**
F15	**J_TDO**	O	–	**Test Data Output** There is a boot strap setting on this pin.This pin is used to decide the interface mode: • 1 = JTAG debugging mode • 0= JTAG test mode
	J_TDO	O		**JTAG Test Data Output**
C13	**J_TMS**	I	–	**Test Mode Select**
	J_TMS	I		**JTAG Mode Select**
D13	**J_TCK**	I	–	**Test Clock**
	J_TCK	I		**JTAG Test Clock**

3.3.11 DDR-SDRAM

Table 56 describes the Double Data Rate (DDR) SDRAM interface.

DDR Ball Names (excl. Power) start with “DDR_”. When the function is unique for LPDDR4, DDR4, or DDR3, only one name is shown after DDR_. When there is a different function for at least one DDR type, the convention after DDR_ is: “LPDDR4_DDR4_DDR3”.

Table 56 DDR-SDRAM Signals

Ball No.	Name	Pin Type	Buffer Type	Function
U46	**DDR_DQ31**	I/O	–	**DDR Data Bus [31:16]**
U45	**DDR_DQ30**	I/O	–	
W47	**DDR_DQ29**	I/O	–	
AF46	**DDR_DQ28**	I/O	–	
AD46	**DDR_DQ27**	I/O	–	
AF47	**DDR_DQ26**	I/O	–	
W45	**DDR_DQ25**	I/O	–	
W46	**DDR_DQ24**	I/O	–	
AR41	**DDR_DQ23**	I/O	–	
AR42	**DDR_DQ22**	I/O	–	
AT42	**DDR_DQ21**	I/O	–	
AT41	**DDR_DQ20**	I/O	–	
Y41	**DDR_DQ19**	I/O	–	
Y42	**DDR_DQ18**	I/O	–	
AB41	**DDR_DQ17**	I/O	–	
AH41	**DDR_DQ16**	I/O	–	
AY46	**DDR_DQ15**	I/O	–	**DDR Data Bus [15:0]**
AV45	**DDR_DQ14**	I/O	–	
AL45	**DDR_DQ13**	I/O	–	
AV46	**DDR_DQ12**	I/O	–	
AN47	**DDR_DQ11**	I/O	–	
AT45	**DDR_DQ10**	I/O	–	
AL46	**DDR_DQ9**	I/O	–	
AY47	**DDR_DQ8**	I/O	–	
BC46	**DDR_DQ7**	I/O	–	
BD47	**DDR_DQ6**	I/O	–	
BC45	**DDR_DQ5**	I/O	–	
BE44	**DDR_DQ4**	I/O	–	
BB45	**DDR_DQ3**	I/O	–	
BF46	**DDR_DQ2**	I/O	–	
BG46	**DDR_DQ1**	I/O	–	
BF47	**DDR_DQ0**	I/O	–	

Table 56 DDR-SDRAM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
BA42	**DDR_ACTN_A15**	O	–	**A15 in DDR4 mode only** DDR CMD ACTIVATE, shared with RAS_N in DDR3 mode
BE40	**DDR_CKAC_BG1_A14**	O	–	**DDR Address Bus [14:0]**
BG24	**DDR_CAB0_A13**	O	–	
BE33	**DDR_CAA0_A12**	O	–	
BF33	**DDR_CAA1_A11**	O	–	
BG44	**DDR_CAB2_A10**	O	–	
BB35	**DDR_A9**	O	–	
BE38	**DDR_CAA3_A8**	O	–	
BF38	**DDR_CAA2_A7**	O	–	
BG38	**DDR_CAA4_A6**	O	–	
BF40	**DDR_CAA5_A5**	O	–	
BB42	**DDR_CAA6_A4**	O	–	
BF31	**DDR_CAA7_A3**	O	–	
BE31	**DDR_CKBT_A2**	O	–	
BE29	**DDR_CKBC_A1**	O	–	
BG43	**DDR_CAB3_A0**	O	–	
BD43	**DDR_CAB6_WEN**	O	–	**DDR Write Enable; DDR4 A14**
BE20	**DDR_CAB7_RASN**	O	–	**DDR Row Address Strobe;DDR4 A16**
BE43	**DDR_CAB5_CASN**	O	–	**DDR Column Address Strobe;DDR4 A15**
BA41	**DDR_CAB8_ODT0**	O	–	**DDR UDM, LDM, ODT in DDR4 modes, DM[3:0] are shared with DBI_n[3:0]**
AY45	**DDR_DM1**	I/O	–	**DDR Data Mask 1**
BD46	**DDR_DM0**	I/O	–	**DDR Data Mask 0**
BF17	**DDR_CAB9_ODT1**	O	–	
AJ45	**DDR_DM2**	I/O	–	**DDR Data Mask 2**
AF45	**DDR_DM3**	I/O	–	**DDR Data Mask 3**
AD45	**DDR_DQS3P**	I/O	–	**Differential DQS Pair for upper DQ lines [DQ24 - DQ31]**
AB45	**DDR_DQS3N**	I/O	–	
AJ41	**DDR_DQS2P**	I/O	–	**Differential DQS Pair for lower DQ lines [DQ16- DQ23]**
AJ42	**DDR_DQS2N**	I/O	–	
AN45	**DDR_DQS1P**	I/O	–	**Differential DQS Pair for upper DQ lines [DQ8 - DQ15]**
AN46	**DDR_DQS1N**	I/O	–	

Table 56 DDR-SDRAM Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
BD45	**DDR_DQS0P**	I/O	–	**Differential DQS Pair for lower DQ lines [DQ0 - DQ7]**
BE45	**DDR_DQS0N**	I/O	–	
BE41	**DDR_CKAT_BG0_BA2**	O	–	**DDR Bank/Group Address [1:0]; BG[0] is BA[2] for DDR3**
BA36	**DDR_CAB1_BA0**	O	-	**DDR BANK 0**
BE22	**DDR_BA1**	O	-	**DDR BANK 1**
BG31	**DDR_CSA0_CSN0**	O	–	**LPDDR Ch A CS0/DDR Chip Select 0**
BG17	**DDR_ODTB_CSN1**	O	–	**LPDDR3 ODTB/DDR Chip Select 1**
BF19	**DDR_CAA8_CK0P**	O	–	**DDR Clock 0 Output**
BE19	**DDR_CAA9_CK0N**	O	–	**DDR Clock 0 Inverted**
BE26	**DDR_CKEB0_CK1P**	O	–	**DDR Clock 1 Output**
BF26	**DDR_CKEB1_CK1N**	O	–	**DDR Clock 1 Inverted**
BE35	**DDR_CKEA0_CKE0**	O	–	**DDR Clock Enable 0**
BE24	**DDR_ZQ**	AI	–	**DDR Calibration Resistor, 120 Ω, 1%**
BB41	**DDR_RST**	O	–	**DDR RESET**
BB36	**DDR_CAPAR**	O	–	**DDR4 only; DDR Parity**
BF24	**DDR_ALTN**	O	–	**DDR4 only; DDR4 ALERT_N**
BA29	**DDR_CKEA1_CKE1**	O	–	**DDR Clock Enable 1**
BB20	**DDR_ODTA**	O	–	**ODT, not used in the system board**
BA35	**DDR_CSA1_C0**	O	–	**LPDDR Ch A CS1/DDR logic Channel Chip Select 0**
BB28	**DDR_CSB0**	O	–	**LPDDR Ch B Chip Select 0**
BA28	**DDR_CSB1**	O	–	**LPDDR Ch B Chip Select 1**
BF43	**DDR_CAB4**	O	–	**LPDDR Command and Address Bus bit 4, Channel B**
BB22	**DDR_C1**	O	–	**DDR4 logic channel Chip Select 1**
BA22	**DDR_MTEST**	AIO	–	**DDR MTEST**

3.3.12 Clock

Table 57 describes the system clock signals.

Table 57 Clock Signals

Ball No.	Name	Pin Type	Buffer Type	Function
B1	**XI**	AI	A	**External Crystal Input/Output**
C1	**XO**	AO	A	
A38	**CLK_1588_N**	AO	A	**CLK 1588 Differential Reference Clock** These pins provide the 25/40 MHz differential reference clock to be provided for clock slave device.
A40	**CLK_1588_P**	AO	A	
R44	**PCIE10_CLKP**	AO	A	**PCIE10 Differential Reference Clock for RC Mode** These pins provide the 100 MHz differential reference clock to be provided for PCIE EP.
N44	**PCIE10_CLKN**	AO	A	
H44	**PCIE11_CLKP**	AO	A	**PCIE11 Differential Reference Clock for RC Mode** These pins provide the 100 MHz differential reference clock to be provided for PCIE EP.
G44	**PCIE11_CLKN**	AO	A	
C28	**PCIE20_CLKP**	AO	A	**PCIE20 Differential Reference Clock for RC Mode** These pins provide the 100 MHz differential reference clock to be provided for PCIE EP.
C26	**PCIE20_CLKN**	AO	A	
A17	**PCIE21_CLKP**	AO	A	**PCIE21 Differential Reference Clock for RC Mode** These pins provide the 100 MHz differential reference clock to be provided for PCIE EP.
A19	**PCIE21_CLKN**	AO	A	

3.3.13 Reset

Table 58 describes the reset signals.

Table 58 Reset Signals

Ball No.	Name	Pin Type	Buffer Type	Function
C7	**POR_N**	I	PU	**Power-on Reset** *Note: Active low*
C8	**HRST_N**	O	OD	**Hardware Reset/Power-on Reset Output** This pin is the POR module output and synchronous with the on-chip reset after power-on. *Note: Active low*

3.3.14 Power

Chapter 15 Electrical Characteristics of URX851/URX850/MxL25641 describes the power supply characteristics.

Table 59 Analog and IO Power Supply

Ball No.	Name	Pin Type	Buffer Type	Function
XO POR, PLL and JTAG Supply				
AA11, AC11	**VDDA1V8OTPPOREMMC**	PWR	–	**1.8 V Power for OTP, PVT, XO, POR, EMMC** Source to OTP memory programming, PVT sensors, XO, POR and EMMC PHY.
T39	**VDDA1V8DDRPLL**	PWR	–	**1.8 V Power for DDR PLL**
F13, G13	**VDDA1V8ROLCPLL**	PWR	–	**1.8 V Power for PLL0/1/2, Low Jitter PLL**
AK9	**VDDD0V8CPULCREF**	PWR	–	**0.80 V Digital Power** CPU and Low Jitter PLL REF Circuit Power supply.
AM9	**VDDD0V8CPULCPOST**	PWR	–	**0.80 V Digital Power** CPU and Low Jitter PLL Post Divider power supply.
AE9	**VDDA1V8CPUPLL**	PWR	–	**1.8 V Power for CPU PLL**
AG11	**DVDD1V8_JTAG**	PWR	–	**1.8 V Power JTAG**
HSIO SerDes (PCI Express, PON, XFI) Supply				
F41, F42	**VPH_1V8_PCIE10_11**	PWR	–	**1.8 V Power for HSIO 10_11**
F8, G8	**VPH_1V8_PCIE20_21**	PWR	–	**1.8 V Power for HSIO 20_21**
D41, D42	**VPH_1V8_PON**	PWR	–	**1.8 V Power PON HSIO 4**
J21, J23	**VPH_1V8_USB1_XFI5**	PWR	–	**1.8 V Power XFI 5 and USB1 HSIO**
AD1	**VDDA1V8CML_LEFT**	PWR	–	**1.8 V Power for CML Analog Module of LCPLL Ref Clk**
R42	**VDDA1V8CML_RIGHT**	PWR	–	**1.8 V Power for CML Analog Module of LCPLL Ref Clk**
USB Supply				
G29	**VDDA3v3_USB2.1**	PWR	–	**3.3 V Power for USB 2.0 PHY Analog of USB 1**
DDR Supply				
AB42, AB44, AH42, AH44, AM30, AM34, AM39, AP32, AP37, AT44, AU30, AU34, AU39, AW32, AW37, BA44, BD22, BD28, BD35, BD41, Y45	**VDDA1V1DDR**	PWR	–	**Power for DDR LPDDR4/DDR4/DDR3L** Supplied with different voltage depending on DDR mode: • 1.1 V for LPDDR4/4x • 1.2 V for DDR4 • 1.35 V for DDR3L
AP39	**VDD_VREF**	PWR	–	**Power for DDR Ref Voltage**

Table 59 Analog and IO Power Supply (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
VDDP GPIO Supply				
AW11, AW9, BA6, BA7, BA8, F6, G6, G7	**VDDD3V3GPIO**	PWR	–	**3.3 V Power for GPIO** Supply to 3.3 V only in System

Table 60 Core Power Supply

Ball No.	Name	Pin Type	Buffer Type	Function
ROC, ADP, JTAG and CPU Digital Supply				
AA14, AA18, AA23, AA27, AA32, AA37, AC16, AC21, AC25, AC30, AC34, AC39, AE14, AE18, AE23, AE27, AE32, AE37, AG16, AG21, AG25, AG30, AG34, AG39, AK14, AK37, AM11, AU11, T27, T32, T37, V30, V34	**VDDD0V8ROC**	PWR	–	**0.80 V ROC/DDR** RoC Domain incl. DDR Core supply. Fixed supply rail
L14, L18, L23, P16, P21, P25, T14, T18, T23, V16, V21, V25	**VDDD0V8ADP**	PWR	–	**0.8 V ADP** Power for ADP domain core supply supporting DVS
AK18, AK23, AM16, AM21, AM25, AP14, AP18, AP23, AU16, AU21, AU25, AW18, AW23	**VDDD0V8CPU0L2**	PWR	–	**0.80 V CPU** L2 and CPU module 0 domain power supply supporting DVS
HSIO SerDes (PCI Express, PON, XFI) Supply				
J34, J37, J39, L34	**VA_0V8_PCIE10_11**	PWR	–	**0.80 V Power for HSIO of Module 10_11**
J11, J9, L9, P9	**VA_0V8_PCIE20_21**	PWR	–	**0.80 V Power for HSIO of Module 20_21**
F35, F36	**VA_0V8_PON**	PWR	–	**0.80 V Power for HSIO of Module 4**
J27, J30, L30, L27	**VA_0V8_USB1_XFI5**	PWR	–	**0.80 V Power for USB1 and XFI5 HSIO**
T11	**VDDA0V8CML_LEFT**	PWR	–	**0.80 V Power for CML Analog Module of LCPLL Ref Clk**

Table 60 Core Power Supply (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
N42	**VDDA0V8CML_RIGHT**	PWR	–	**0.80 V Power for CML Analog Module of LCPLL Ref Clk**
DDQ eMMC Supply				
AW14	**VDDA0V8VCEMMC**	PWR	–	**0.8 V Analog Power for EMMC DLL Supply**
Sense				
N4	**M0_SENSE_P**	PWR	–	**Voltage Sensing for CPU Module 0**
R4	**ADP_SENSE_P**	PWR	–	**Voltage Sensing for ADP**
AJ7	**ROC_SENSE_P**	PWR	–	**Voltage Sensing for RoC**

3.3.15 Power / Ground / Test

Chapter 15 Electrical Characteristics of URX851/URX850/MxL25641 describes the power / ground supply.

Table 61 Power Supply Ground & Test Signals

Ball No.	Name	Pin Type	Buffer Type	Function
Ground Supply				
A26, A31, A47, AC9, B17, B19, B26, B38, B40, C19, C24, C29, C33, C35, C40, C43, C45, C46, D20, D28, D29, D35, D36, E46, F20, F22, F46, G15, G20, G22, G28, G35, G42, H42, H45, H6, H7, J14, J16, J18, J25, J32, K45, L11, L32, L37, L39, M47, N41, N45, P11, P27, P30, P32, P34, P37, P39, R41, R45, V11	**VSSA**	GND	–	**Ground**
A6, AA16, AA21, AA25, AA30, AA34, AA39, AC14, AC18, AC23, AC27, AC32, AC37, AD47, AE11, AE16, AE21, AE25, AE30, AE34, AE39, AG14, AG18, AG23, AG27, AG32, AG37, AH45, AJ44, AK16, AK21, AK25, AK39, AL1, AL47, AM14, AM18, AM23, AM32, AM37, AP11, AP16, AP21, AP25, AP30, AP34, AR44, AR45, AT6, AU14, AU18, AU23, AU32, AU37, AV1, AV47, AW16, AW21, AW25, AW30, AW34, AW39, B2, BA45, BB29, BB44, BB6, BB7, BC1, BC47, BD15, BD20, BD29, BD36, BD42, BD7, BE17, BE28, BE36, BE42, BF44, BG1, BG12, BG19, BG26, BG33, BG40, BG47, BG6, C2, D7, E1, F7, H4, K1, L16, L21, L25, P14, P18, P23, T16, T21, T25, T30, T34, U1, U47, V14, V18, V23, V27, V32, V37, V39, Y44	**VSSP**	GND	–	**Ground for PAD** Ground for digital core, I/O logic
T9	**TEST26**	O	--	**TEST PIN 26** Leave not connected (NC) in system board.
AN2	**TEST40**	O	--	**TEST PIN 40** Leave not connected (NC) in system board.
AL2	**TEST43**	O	--	**TEST PIN 43** Leave not connected (NC) in system board.
K3	**TEST76**	O	--	**TEST PIN 76** Leave not connected (NC) in system board.
M2	**TEST78**	O	--	**TEST PIN 78** Leave not connected (NC) in system board.

Table 61 Power Supply Ground & Test Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
M3	**TEST79**	O	--	**TEST PIN 79** Leave not connected (NC) in system board.
R3	**TEST80**	O	--	**TEST PIN 80** Leave not connected (NC) in system board.
N3	**TEST81**	O	--	**TEST PIN 81** Leave not connected (NC) in system board.
AJ6	**TEST87**	O	--	**TEST PIN 87** Leave not connected (NC) in system board.
F2	**TEST88**	O	--	**TEST PIN 88** Leave not connected (NC) is system board.
E3	**TEST89**	O	--	**TEST PIN 89** Leave not connected (NC) in system board.
H3	**TEST99**	O	--	**TEST PIN 99** Leave not connected (NC) in system board.
AK30	**TEST_DQP**	O	--	**TEST PIN DQP** Leave not connected (NC) in system board.
AK32	**TEST_DQN**	O	--	**TEST PIN DQP** Leave not connected (NC) in system board.
BA15	**TEST_DM**	O	--	**TEST PIN DM** Leave not connected (NC) in system board.
AW27	**TEST_D7**	O	--	**TEST PIN D7** Leave not connected (NC) in system board.
AK34	**TEST_D6**	O	--	**TEST PIN D6** Leave not connected (NC) in system board.
AP27	**TEST_D5**	O	--	**TEST PIN D5** Leave not connected (NC) in system board.
AU27	**TEST_D4**	O	--	**TEST PIN D4** Leave not connected (NC) in system board.

Table 61 Power Supply Ground & Test Signals (cont'd)

Ball No.	Name	Pin Type	Buffer Type	Function
AK27	**TEST_D3**	O	--	**TEST PIN D3** Leave not connected (NC) in system board.
BB15	**TEST_D2**	O	--	**TEST PIN D2** Leave not connected (NC) in system board.
BA20	**TEST_D1**	O	--	**TEST PIN D1** Leave not connected (NC) in system board.
AM27	**TEST_D0**	O	--	**TEST PIN D0** Leave not connected (NC) in system board.

3.3.16 Fuse

Table 62 describes how the fuse pins must be connected.

Table 62 Fuse Signals

Ball No.	Name	Pin Type	Buffer Type	Function
AG9	**VDDA1V8FUSE**	I	–	**1V8 Fuse Power Supply**

3.4 Digital Signal Ball (Pin) Reset Property and Drive Strength

This section describes the reset values and drive strength of the digital IO pins.

Table 63 defines the abbreviations used in this section.

- The **Reset State** is the logic level of a pin during the POR phase (internal Power_on Reset signal in Figure 15 is low).
- The **Reset Release State** is the logic level of a pin after the POR signal (internal Power_on Reset signal in Figure 15) transition from low to high.

Table 63 Abbreviations for the Reset Property

Abbreviation	Description
Z	High impedance
L	Pin not driven, internal pull-down
H	Pin not driven, internal pull-up

3.4.1 TDM

Table 53 describes the TDM interface multiplexing.

Table 64 TDM Signals

Name	Reset State	Reset Release State	Drive Strength (mA)	Voltage Domain
C55_TDM1_DCL	L	Z	4	3.3 V
C55_TDM1_DI	L	Z	4	3.3 V
C55_TDM1_DO	L	Z	4	3.3 V
C55_TDM1_FSC	L	Z	4	3.3 V
C55_TDM0_DO	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
C55_TDM0_FSC	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V

3.4.2 8-bit NAND Flash

Table 65 describes the 8-bit NAND Flash interface.

Table 65 NAND Interface Signals

Name	Reset State	Reset Release State	Drive Strength (mA)	Voltage Domain
GPIO50	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO51	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO52	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO53	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO54	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO55	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO56	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO57	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO23	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO49	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO13	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO24	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO48	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO59	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO60	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO61	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V

3.4.3 Serial Peripheral Interface (SPI)

The SPI interface is targeted for serial flash support.

Table 66 SPI/GPIO Interface Signals

Name	Reset State	Reset Release State	Drive Strength (mA)	Voltage Domain
GPIO10	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO11	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO14	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO19	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO15	L	Z[1)]	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO16	L	Z[2)]	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO18	L	Z[3)]	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO17	L	Z[4)]	2, programmable (2, 4, 8, 12 mA)	3.3 V

1) ROM programs it to SPI0_CS1_N when boot strapping selects SPI. In this case, the ROM programs the pin according to the needs of the SPI master interface.
2) ROM programs it to SPI0_RX when boot strapping selects SPI. In this case, the ROM programs the pin according to the needs of the SPI master interface.
3) ROM programs it to SPI0_CLK when boot strapping selects SPI. In this case, the ROM programs the pin according to the needs of the SPI master interface.
4) ROM programs it to SPI0_TX when boot strapping selects SPI. In this case, the ROM programs the pin according to the needs of the SPI master interface.

3.4.4 UART

Several peripherals can be connected to the UART. One UART interface is intended for debugging (OS console) purposes when console on UART is enabled.

Table 67 UART Interface Signals

Name	Reset State	Reset Release State	Drive Strength (mA)	Voltage Domain
UART1_RX	Z	Z	2 mA	3.3 V
UART1_TX	Z	L	2 mA	3.3 V
UART0_RX	Z	Z	2 mA	3.3 V
UART0_TX	Z	L	2 mA	3.3 V
UART2_RX	Z	Z	2 mA	3.3 V
UART2_TX	Z	L	2 mA	3.3 V
UART3_RX	Z	Z	2 mA	3.3 V
UART3_TX	Z	L	2 mA	3.3 V

3.4.5 Peripherals, General Purpose I/O, Reset

This section describes the peripherals and general purpose I/O pins. For all other GPIO pins not mentioned in **Section 3.4.1**, **Section 3.4.2**, **Section 3.4.3**, **Section 3.4.4**, follow the settings given in **Table 68**.

Table 68 GPIO Interface Signals

Name	Reset State	Reset Release State	Drive Strength (mA)	Voltage Domain
GPIO0	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO1	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO2	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO3	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO4	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO5	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO6	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO7	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO8	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO9	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO21	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO22	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V
GPIO42	L	Z	2, programmable (2, 4, 8, 12 mA)	3.3 V

4 CPU Subsystem

The CPU subsystem chapter covers these sections:

4.1 Intel Atom CPU Subsystem

This section describes the Intel Atom CPU Subsystem.

4.1.1 Overview

The CPU subsystem consists of Intel Atom cores, caches, and the system agent which connects these modules together.

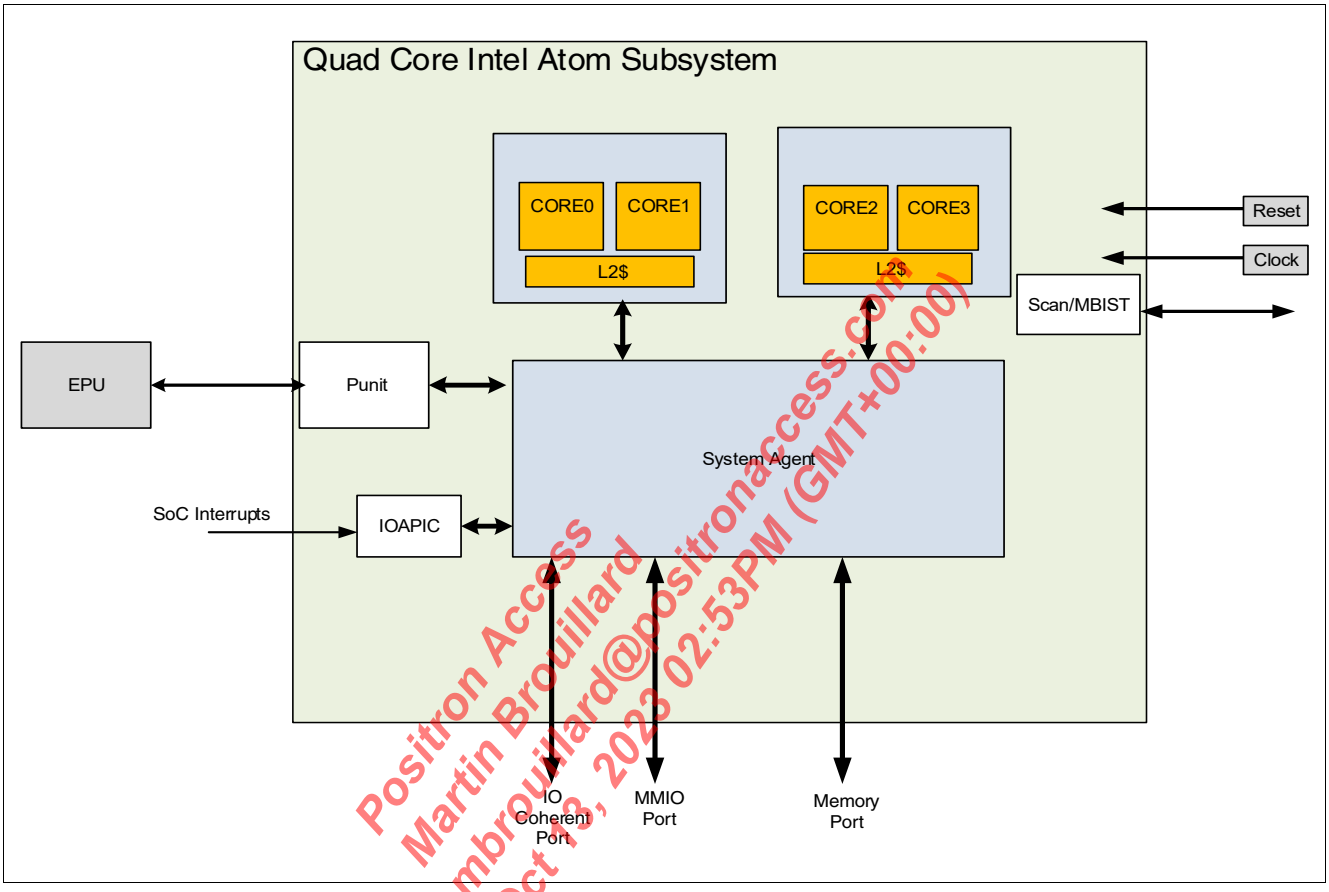


Figure 7 Quad Core CPU Subsystem Overview

Table 69 shows the quad core CPU subsystem address maps.

Table 69 Quad Core CPU Subsystem

Subsystem	Sub-module	Size (Hex)	Size	CPU Physical Address Start (Hex)	CPU Physical Address End (Hex)	URX851/URX850 Decoded to
MC	MC memory	4000_0000	1024 MB	0_0000_0000	0_3FFF_FFFF	Memory Port Range 0
	MC memory	4000_0000	1024 MB	0_4000_0000	0_7FFF_FFFF	Memory PortRange 0
	MC memory	4000_0000	1024 MB	0_8000_0000	0_BFFF_FFFF	Memory PortRange 0
MMIO	MMIO	1000_0000	256 MB	0_C000_0000	0_CFFF_FFFF	MMIO Range 0
	MMIO	1000_0000	256 MB	0_D000_0000	0_DFFF_FFFF	MMIO Range 1
	MMIO	1000_0000	256 MB	0_E000_0000	0_EFFF_FFFF	MMIO Range 2
	MMIO	800_0000	128 MB	0_F000_0000	0_F7FF_FFFF	MMIO Range 3
APIC	IO-APIC	10_0000	1 MB	FEC0_0000	FECF_FFFF	IA - Internal
	Reserved	10_0000	1 MB	FED0_0000	FEDF_FFFF	IA - Internal
	LAPIC	1000	4 MB	FEE0_0000	FEE0_0FFF	IA - Internal
	Reserved	F_F000	1020 MB	FEE0_1000	FEEF_FFFF	IA - Internal
	Reserved	100_0000	16 MB	FEF0_0000	FFEF_FFFF	IA - Internal
MMIO	MMIO (Boot ROM)	10_0000	1 MB	FFF0_0000	FFFF_FFFF	MMIO Range 4
MC	MC memory	40000000	1024 MB	1_0000_0000	1_3FFF_FFFF	Memory PortRange 0
	MC memory	40000000	1024 MB	1_4000_0000	1_7FFF_FFFF	Memory PortRange 0
	MC memory	40000000	1024 MB	1_8000_0000	1_BFFF_FFFF	Memory PortRange 0
	MC memory	40000000	1024 MB	1_C000_0000	1_FFFF_FFFF	Memory PortRange 0
	MC memory	40000000	1024 MB	2_0000_0000	2_3FFF_FFFF	Memory PortRange 0
	MC memory	40000000	1024 MB	2_4000_0000	2_7FFF_FFFF	Memory PortRange 0
	MC memory	40000000	1024 MB	2_8000_0000	2_BFFF_FFFF	Memory PortRange 0
	MC memory	40000000	1024 MB	2_C000_0000	2_FFFF_FFFF	Memory PortRange 0
	MC memory	40000000	1024 MB	3_0000_0000	3_3FFF_FFFF	Memory PortRange 0
	MC memory	40000000	1024 MB	3_4000_0000	3_7FFF_FFFF	Memory PortRange 0
	MC memory	40000000	1024 MB	3_8000_0000	3_BFFF_FFFF	Memory PortRange 0
	MC memory	40000000	1024 MB	3_C000_0000	3_FFFF_FFFF	Memory PortRange 0
	MC memory	40000000	1024 MB	4_0000_0000	4_3FFF_FFFF	Memory PortRange 0

4.1.2 Features

The CPU subsystem is a cluster of processors interconnected to enable cache coherence in the primary and secondary data caches, as well as IO system masters. The Intel Atom subsystem consists of these units:

- For URX851/URX850
  - 4x Intel Atom Cores in two modules, each module has two cores
  - Four cores are in identical configuration
  - L2$ Controller (L2$), per dual core module sharing L2$ memory
  - L2$ 1 MB per module
- For MxL25641
  - 2x Intel Atom Cores in one modules
  - 2 cores are in identical configuration
  - L2$ Controller (L2$), dual core module sharing L2$ memory
  - L2$ 1 MB
- Common for all products
  - Cache coherence System Agent (SA)
  - IO coherent path to allow system masters to access the Intel Atom subsystem caches in a coherent way
  - Up to 256 bit interrupt controller (IOAPIC), 32 reserved for the Intel Atom subsystem usage
  - P-unit power controller for the cores, Dynamic Voltage and Frequency Scaling (DVFS) management
  - Debug and trace system

4.1.2.1 Intel Atom Processor Core

The Airmont class Intel Atom processor core provides these features:

- Intel 64 and IA-32 architecture
- 2-wide instruction decode and out of order execution
- Intel Virtualization technology (Vt-x, Vt-x2 with extended page tables)
- MaxLinear Advanced Encryption New Instructions (AES-NI)
- SSE4.1/4.2 new instructions for media data process acceleration
- Instruction RDRAND and underlying Digital Random Number Generator (DRNG) hardware implemented. The DRNG uses the RDRAND instruction to generate high-quality keys for cryptographic protocols.
- Clock Local Power Controller (CLPU) per core, working with P-unit for MaxLinear Enhanced Deeper Sleep, Deep Power Down, Smart Idle, Smart Power technologies
- L1 Cache
  - 32 KB Instruction cache, 8 way associative
  - 24 KB data cache, 6 way associative
  - Hardware prevention of L1$ aliasing
- Translation Look-aside Buffer (TLB)
  - 32 entries micro TLB, 4 KB per page
  - 48 entries instruction TLB, 4 KB per page
  - 256 entries small page size (4 KB) data TLB
  - 16 entires large page size (2 M/4 MB) data TLB

4.1.2.2 System Agent

The SA supports:

- Coherent link to the CPU core modules via the Inter-Die-Interface (IDI) bus
  - The B-unit handles the CPU request and other masters, arbitrates the requests, and schedules the requests
  - The T-unit tracks all requests received and is responsible for the coherence protocol (snooping and collecting replies)
- MMIO bus interconnecting to the CPU core via A-unit
- DDR memory access bus interconnecting to the CPU core

- IO coherent path for system masters with Isolated Memory Region (IMR) for access restriction
- Addressing decoder
- Interrupt delivery
- Security access: SA (Security attributes of Initiator) generation and delivery, access control for modules inside subsystem
- Power management via P-Unit
- Core initialization and reset control

4.1.2.3 P-Unit

The P-Unit is responsible for processor power management and control functions for the CPU subsystem.

P-Unit constantly monitors the status of the IPs it is linked to, and can control their power state either using dedicated signals or through its PMlink interface. For instance, P-Unit controls the CPU C-states (idle modes) and P-states (power-optimized active modes), their operating clock frequencies, power gating for CPU subsystem modules. The P-unit works autonomously.

The P-Unit performs a series of initialization steps to start the CPU frequency setting, voltage setting changes to non-default based on the power state definitions.

4.1.2.4 I/O Advanced Programmable Interrupt Controller

The I/O Advanced Programmable Interrupt Controller (IOAPIC) provides multi-processor interrupt management and incorporates both static and dynamic symmetric interrupt distribution across all processors. In systems with multiple I/O subsystems, each subsystem can have its own set of interrupts. Each interrupt pin is individually programmable as either edge or level triggered. The interrupt vector and interrupt steering information can be specified per interrupt. An indirect register accessing scheme optimizes the memory space needed to access the IOAPIC's internal registers.

4.1.2.5 Virtualization

VT-x represents Intel's technology for virtualization on the x86 platform, it is a kind of hardware assistant virtualization. it is also called VMX, which stands for Virtual Machine Extensions.

VMX adds ten new instructions: VMPTRLD, VMPTRST, VMCLEAR, VMREAD, VMWRITE, VMCALL, VMLAUNCH, VMRESUME, VMXOFF, and VMXON. These instructions permit entering and exiting a virtual execution mode where the guest OS perceives itself as running with full privilege on a logic processor, but the host OS(VMM) remains protected. VMM has full control of the processor(s) and other platform hardware, manages the guest OS virtual machine.

Two VMX operations: VMX root operation and VMX non-root operation. VMM runs the VMX root operation and VM runs in VMX non-root operation. The transition into non-root operation is called VM entries; transition from VMX non-root to root operation is called VM exits.

Processor in VMX root operation is the same as outside VMX operation except new instructions are available now. Processor in VMX non-root operation is restricted and modified to facilitate virtualization. Certain instructions and events cause VM exits.

However, there is no software visible flag to indicate a logical processor is in VMX root or non-root operations. VMM enters VMX operation by executing a VMXON instruction.

Using VM entries, a VMM can enter a guest into virtual machine. the VMM effects a VM entry using instructions VMLAUNCH and VMRESUME.

VM exits transfer control to an entry point specified by the VMM.

VMM could decide to shut itself down and leave VMX operation by executing the VMXOFF instruction.

VT-x2

VT-x2 is the Intel virtualization technology enhancements for memory virtualization that supports address translation: EPT (Extended Page Table) and VPID (Virtual Processor Identifier).

Under VT-x, VMX operation requires VMX transitions to flush the TLBs and paging-structure caches to ensure translations cached for old linear address space would not be used after the transition. With VPID, VMX transitions may retain cached information and the logical processor can switch to a different linear address space, thus improve the performance for VM exits and VM entries.

EPT is a CPU mechanism for remapping guest-physical memory references. It allows guest to retain control of legacy Intel 64 paging, reduces the frequency of VM exits to VMM, and eliminates the need of shadow-page tables resulting in improved performance.

4.2 Clock System

The Clock Generation Unit (CGU) description covers these sections:

- **Features** (**Section 4.2.1**)
- **Functional Description** (**Section 4.2.2**)

4.2.1 Features

The CGU supports:

- Single crystal based clock system, 40 MHz crystal is attached as system default; oscillator clock input is also accepted (single ended or differential)
- Crystal divided clock (20 MHz) for USB subsystems
- Top Level PLLs
  - Fractional PLL0cz with SSC capability running at 2496 MHz for all Intel Atom subsystem uncore clocks (312 MHz) as well as input clocks to PLLs for CPU module 0 and module 1 clocks
  - CPU module 0 and 1 PLL runs at 3120 MHz as default
  - Fractional PLL0b with SSC capability running at 2400 MHz for NoCs and GSWIP-O
  - Fractional PLL-PP with SSC capability running at 2400 MHz for PPV4 Clocks
  - Fractional PLL2 with SSC capability running at 2000 MHz for DDR subsystem clocks
  - Fractional PLL1 running at 3145.728 MHz for voice related clocks
  - Low Jitter PLL4 generates 100 MHz reference clock for high speed IO with SSC capability
  - LJ-PLL5 generates reference clock for WAN SerDes PHY (in WANPHY subsystem) and support loop timing from Rx to Tx.
  - Low jitter PLL3 for SyncE and XFI reference clock generation

4.2.2 Functional Description

The CGU provides all necessary clock frequencies for the system on chip, using a single crystal and multiple PLLs/dividers/muxes to generate the required clocks, as shown in **Figure 8** to **Figure 10**.

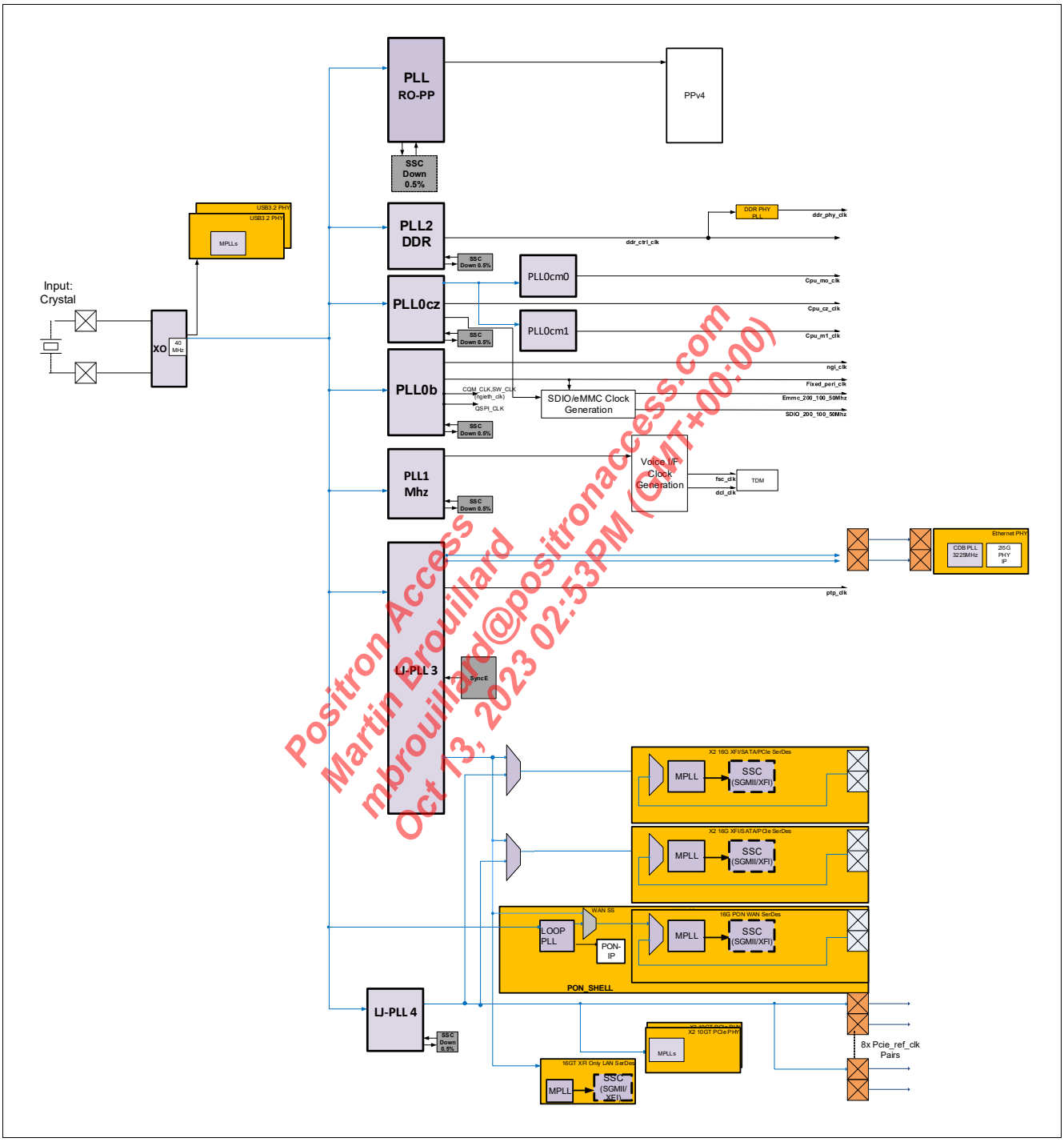


Figure 8 URX851 CGU Overview

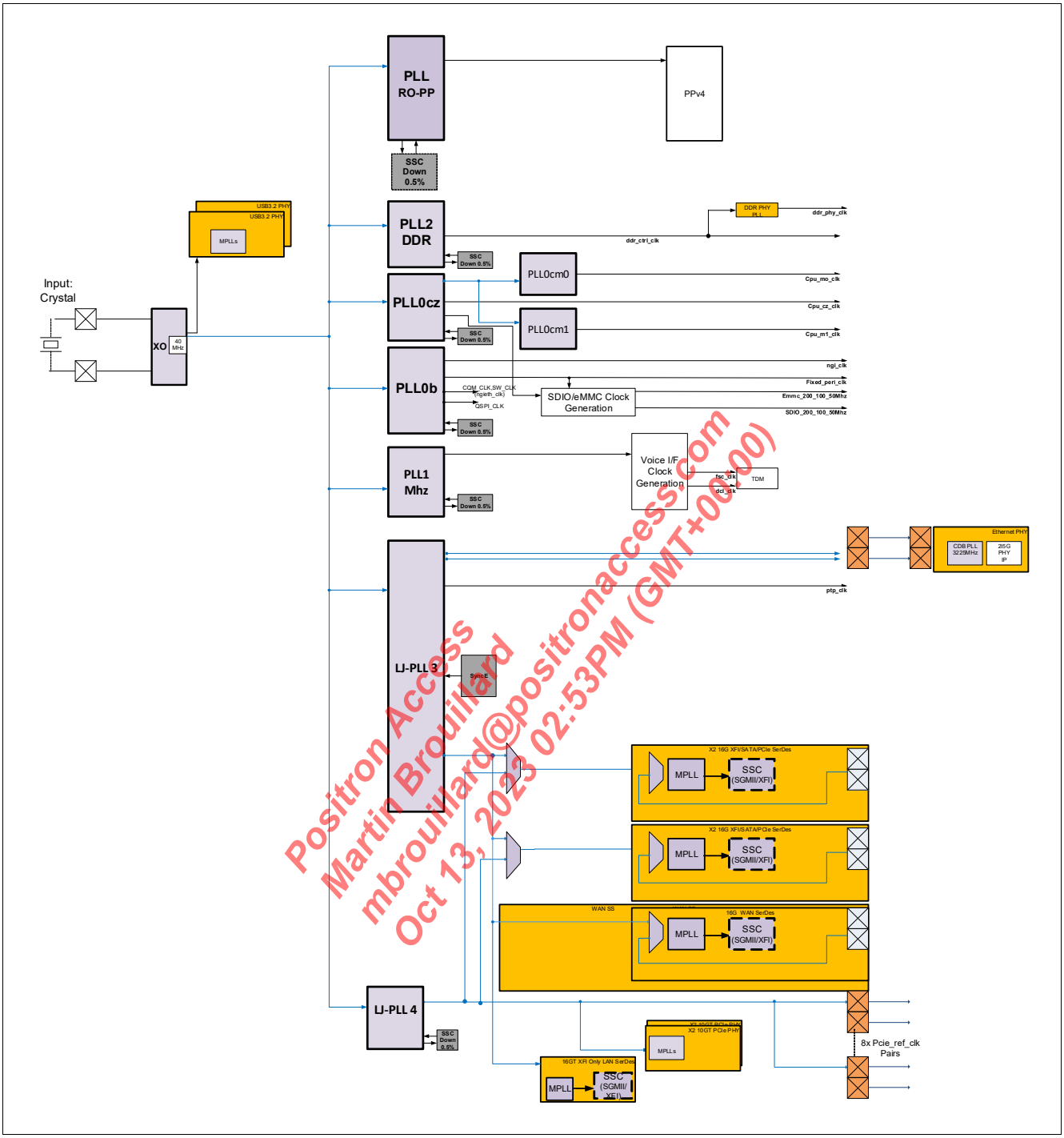


Figure 9 URX850 CGU Overview

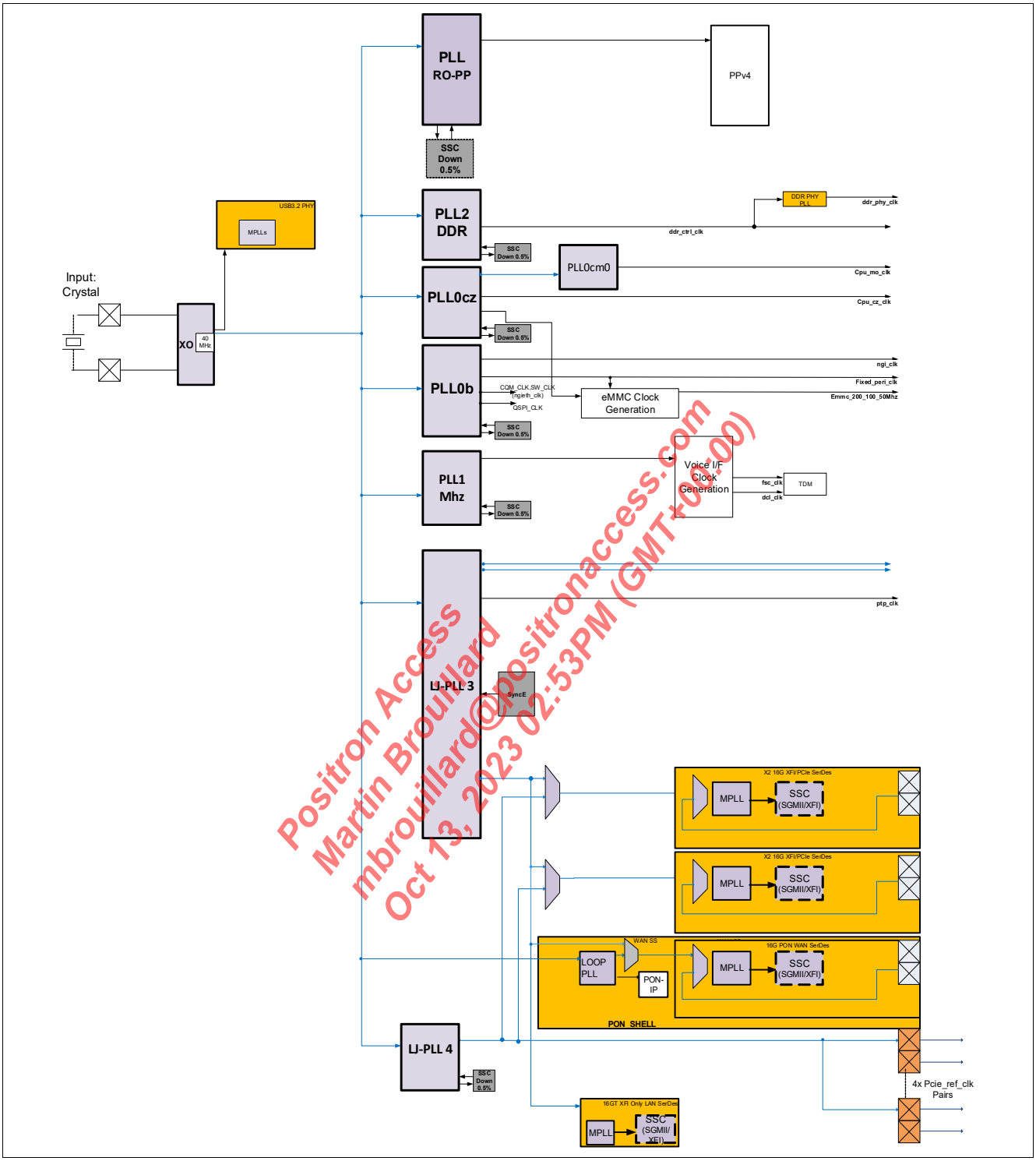


Figure 10 MxL25641 CGU Overview

4.2.3 Clock Input Option and Oscillator Circuit

The oscillator circuit is designed to work either with an external crystal or an external clock source. An oscillator circuit consists of an inverting amplifier with XTAL1 as input and XTAL2 as output. An AC-coupled shaper stage provides 50% duty cycle.

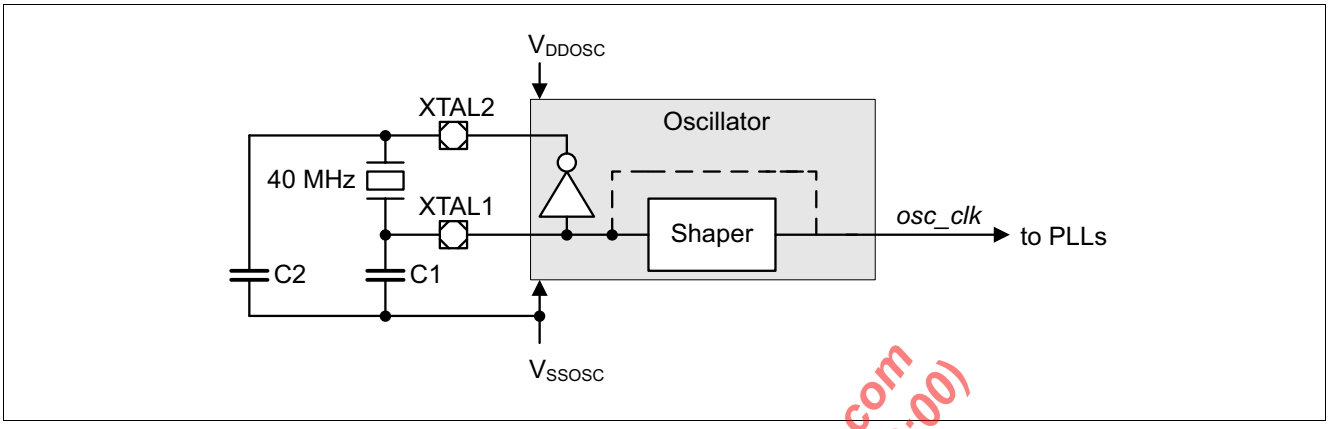


Figure 11 Clock Input Options

4.2.4 External Clock Output or Input Interface

The CGU provides clocks at the external pins used for multiple purposes. The clocks are selected inside the CGU by register **CGU_IF_CLK** and are available as alternative functions at GPIO pins.

These clocks can be selected and provided at the GPIOs:

- SSI_CLKO: 8.192 MHz for connecting to a SmartSLIC. Output mode only.
- GPC1: configurable to be one of the below:
  - 40 MHz output mode: buffered XTAL clock
  - 25 MHz or 10 MHz output mode: this is a 25 MHz or 10 MHz clock output from LJ-PLL3.
  - 1.544 MHz or 2.048 MHz or 8 kHz clock: the clock is divided from LJ-PLL3.
  - Input mode: it is used for clock source of Synchronous Ethernet (SyncE), NTR for voice subsystem.
- GPC2: configurable to be either:
  - 40 MHz output mode: buffered XTAL clock
  - 25 MHz or 10 MHz output mode: this is a 25 MHz or 10 MHz clock output from LJ-PLL3.
  - 1.544 MHz or 2.048 MHz or 8 kHz clock: the clock is divided from LJ-PLL3.
  - Input mode: it is used for clock source of SyncE and also NTR for voice subsystem.
- PPS, configurable to be either:
  - Output mode: low jitter PPS pulse with programmable frequency and width or CPU triggered pulse for reference time synchronization with other chips.
  - Input mode: it is used for reference time synchronization with other chips.

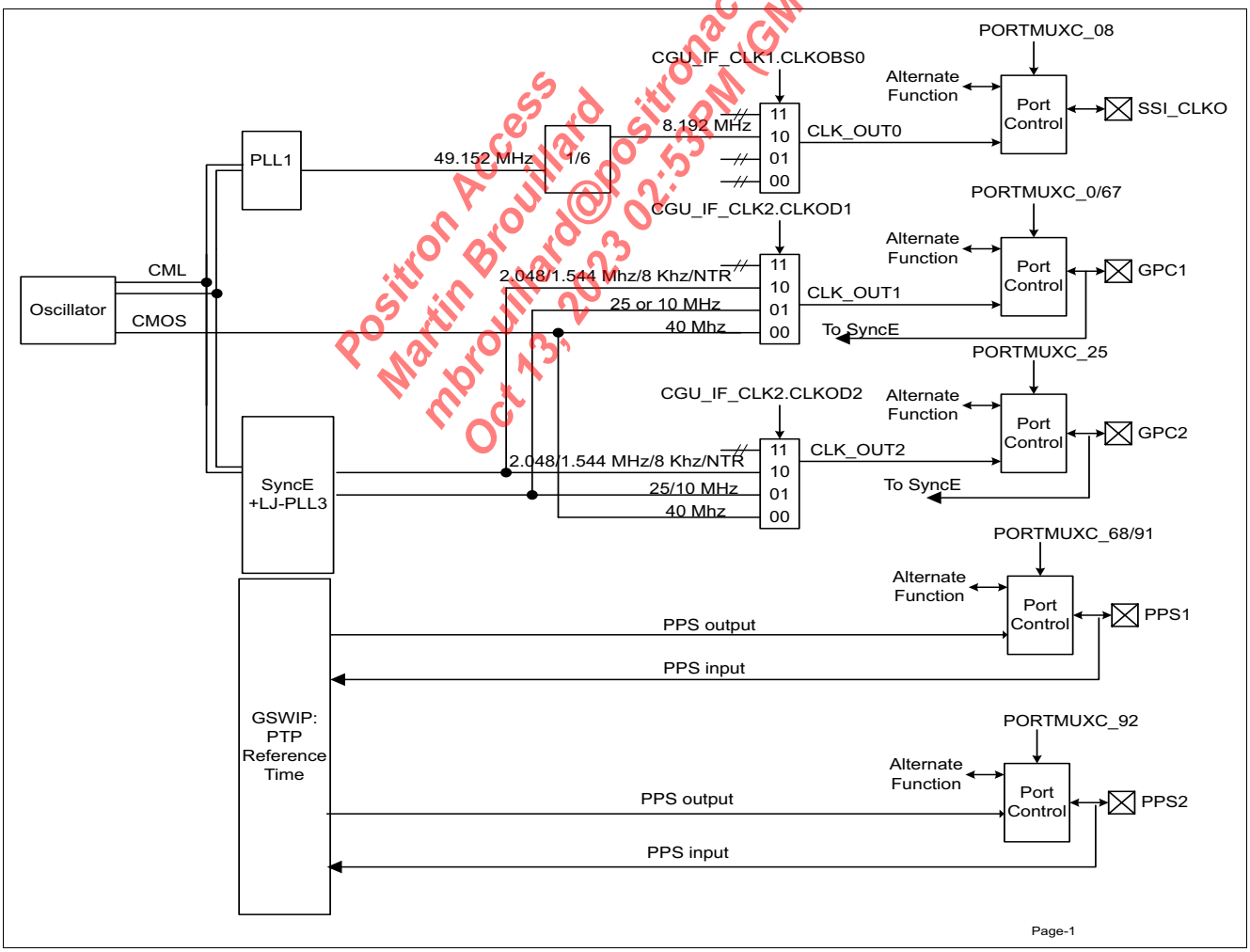


Figure 12 External Clock Output and Input Interface

4.2.5 SyncE

URX851, URX850, and MxL25641 support SyncE.

It is configurable to synchronize to one of these clock sources:

- Recovered received clock by internal SerDes
- Recovered received clock by external 2.5GBASE-T PHY
- Input clock via GPC1 or GPC2
- Programmable bandwidth 0.1 Hz, 3 Hz, and 100 Hz

These functions are clocked from the synchronized clock:

- SerDes in non-PCIe modes
- External 2.5GBASE-T PHY
- 1588/PTP timers
- Output clock via GPC1 or GPC2

Loss of signal detection is integrated on chip and all the modules are supplied seamlessly with clocks generated from the local source when the external master clock signal is not available.

4.2.6 Time of Day

Based on the local real time clock, URX851, URX850, and MxL25641 derive a high precision Pulse Per Second (PPS). This PPS is optionally output on pin PPS and has a configurable active high pulse width. The positive edge of the 1 PPS signal exactly defines the Time-of-Day (ToD) information carried within a message forwarded via the PON subsystem UART1 interface.

4.3 Network-on-Chip Interconnect

This section describes the Network-on-Chip (NoC) interconnect in these sections:

- **Feature List and Conceptual Ideas** (**Section 4.3.1**)
- **Address Map** (**Section 4.3.2**)

4.3.1 Feature List and Conceptual Ideas

The NoC provides these features:

- Transaction-based interconnect architecture
  Native support for transaction level protocols such a AXI, AHB, APB, and OCP
- Handling of tasks
  - Routing transaction from request to target
  - Returning target response to initiator
  - Arbitrating between transactions
  - Implementing QoS policies
  - Logging errors
- Individual transaction initiator and target attached to the NoC via Network Interface Unit (NIUs)
  Each NIU converts the agent protocol to a generic, uniform packet layer. The packets are serialized to narrow links before being routed inside the NoC.
- Maximum average throughput within the interconnect
- Optimal connectivity of the memory system to the interconnect
  - Four times data path ratio between the external DDR SDRAM interface (32 bit double data rate) and the interconnect (256 bit equivalent but serialized after packetization)
- Memory scheduler
  - Implements up to eight ports within the memory scheduler (seven ports configured)
  - Optimizes DRAM efficiency by choosing the best transaction to execute inside each request port's command queue.
  - Arbitrates according to transactions' QoS characteristics
  - Makes trade-off between the QoS requirements and global DDR efficiency.
  - Supports up to 32 DRAM address mapping (map system address to DRAM bank, Row, Column etc bits)
- QoS, bandwidth assignment, and prioritization
- Extra wide read data paths from the DDR SDRAM controller and memory scheduler to the initiator agents
  - Accepts read data at the initiator agent cycle by cycle at the supplied read data rate
  - Does not block the DDR SDRAM controller caused by backpressuring the read data
- Future proven architecture
  - 16 Gbyte DDR SDRAM address range, 1 GB IO address range in-between
  - Global address mapping scheme defaults for IO-coherent architecture, allows flexible switch to non-IOC mapping upon power on early stage configuration
  - Up to 64 Mbyte address range per PCIe controller (memory, configuration, and IO)
- High-performance
  - Supports multiple outstanding read transactions per initiator agent
  - Read data may be returned out of order to minimize the average read latency
  - Point-to-point links between the system master and the interconnect, rather than traditional multi-master-multi-slave bus systems
- Implements security and firewalling functionality inside NoC
- Error capture and logging capabilities per initiator agent and target agent

4.3.2 Address Map

Figure 13 shows the global address mapping.

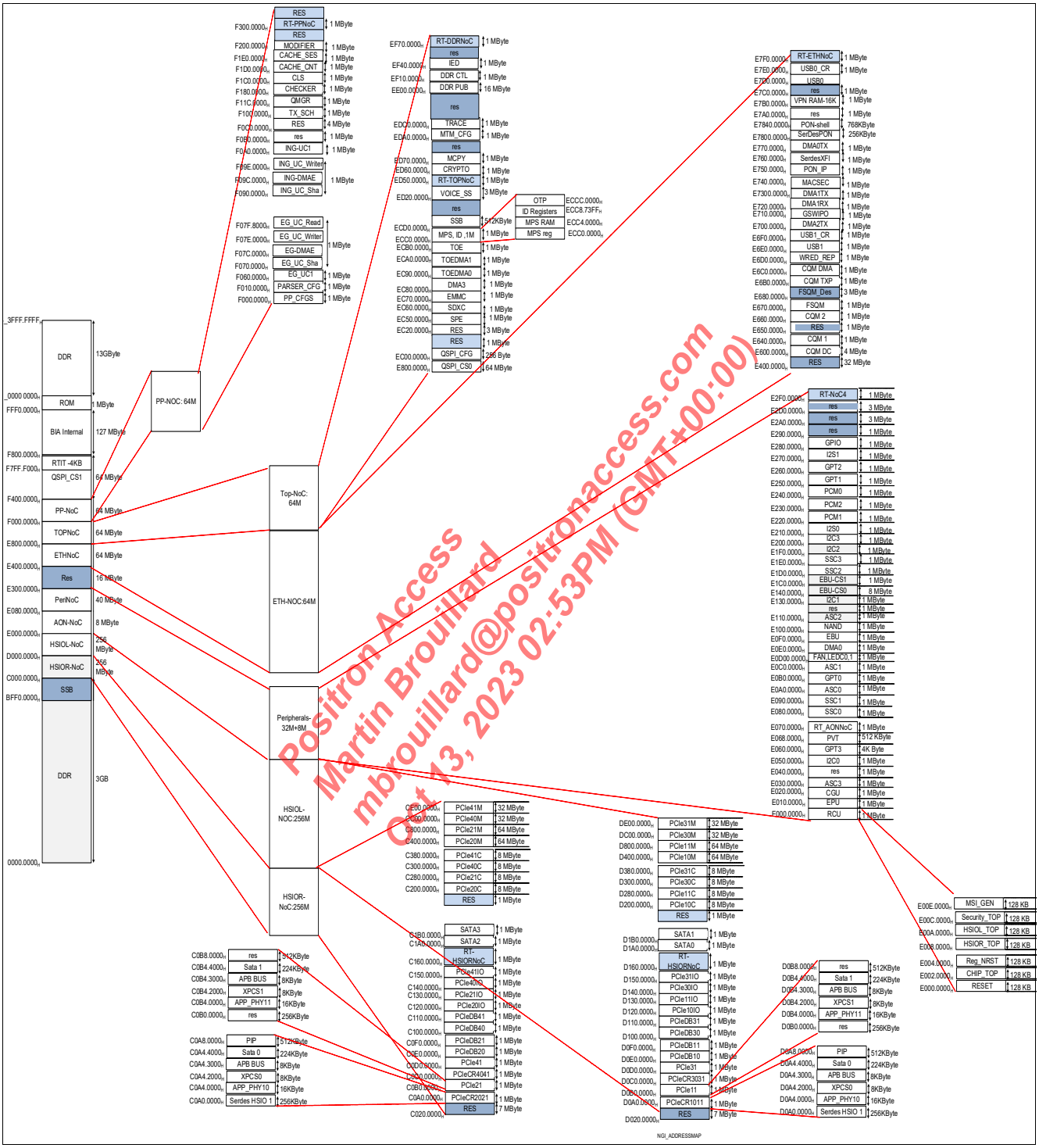


Figure 13 Global Address Map Intel Atom View

Attention: The PON-IP address space is not used in URX850.

Attention: These address spaces are not used in MxL25641: PCIE30/31;PCIE40/41;SATA1/2/3/4 of HSIOL and HSIOR;SDXC;USB0;DDR space 0x2 4000 0000 onwards.

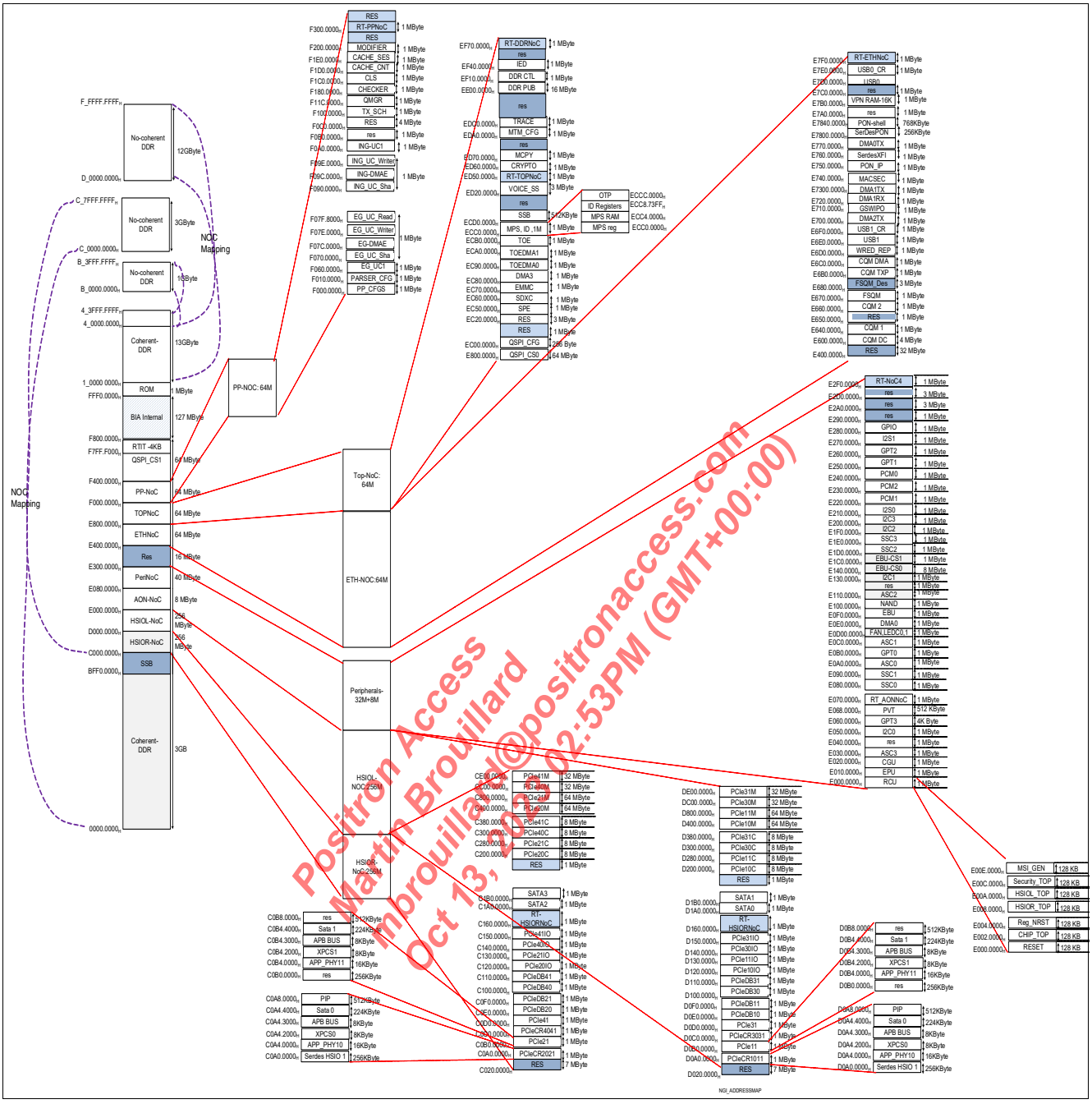


Figure 14 Global Address Map IO Master View

Attention: The PON-IP address space is not used in URX850.

Attention: These address spaces are not used in MxL25641: PCIE30/31;PCIE40/41;SATA1/2/3/4 of HSIOL and HSIOR;SDXC;USB0;DDR space 0x2 4000 0000 onwards.

4.3.2.1 NoC Access Control Security

The NoC access control is done via multiple firewalls inside the NoC.

The firewall takes in runtime input signals, registers defined address region for a specific SAI, and access permission for specific SAI and does region and permission check to grant access on a transaction basis.

The features of the firewalls inside NoC are:

- Each firewall has 64 address regions to be programmed.
- There is a firewall configuration lock feature. Upon the lock, it is not possible to change any firewall configurations, unless a reboot happens.
- Firewalls attached to DDR target port: TEP and peripheral masters to access DDR different regions DMA masters (internal or external) does channel based DDR region separated access.
- Firewalls attached to Shared SRAM Buffer (SSB) target port: address region based isolation for TEP and peripheral masters
- Firewalls inside HSIO NoC: control external master access to SoC internal resource
- Firewalls inside TOPNOC: picked based on full peripherals IO access path to cover all peripherals except DDR and SSB

4.4 Reset System

The reset system description covers these sections:

- **Features** (**Section 4.4.1**)
- **Functional Description** (**Section 4.4.2**)

4.4.1 Features

The reset system supports:

- Power-On Reset (POR) generation
  - POR Input Pin
  - UVD
- Software reset requests and reset domains
- Watchdog timer reset request and reset domain
- Reset Control Unit (RCU) with finite state machine controlling the reset flow

4.4.2 Functional Description

The reset system mainly consists of two parts:

- **Section 4.4.2.1** describes the POR logic.
- **Section 4.4.2.2** describes the RCU.

The POR logic controls also the pin strapping functionality. Additionally, the reset system provides a reset output pin, depicted in **Figure 15**. Finally, the system on chip implements components, which can trigger the RCU to issue various resets.

4.4.2.1 Overview

Figure 15 gives an overview of the reset system. There are multiple power domains inside the SoC digital core supply. However, the monitored domain is the Always ON (AON) domain, part of ROC domain. The other domains (CPU power rails ADP rail) are controlled by EPU to give power good indications. Only when the AON domain power is stable, POR reset is released.

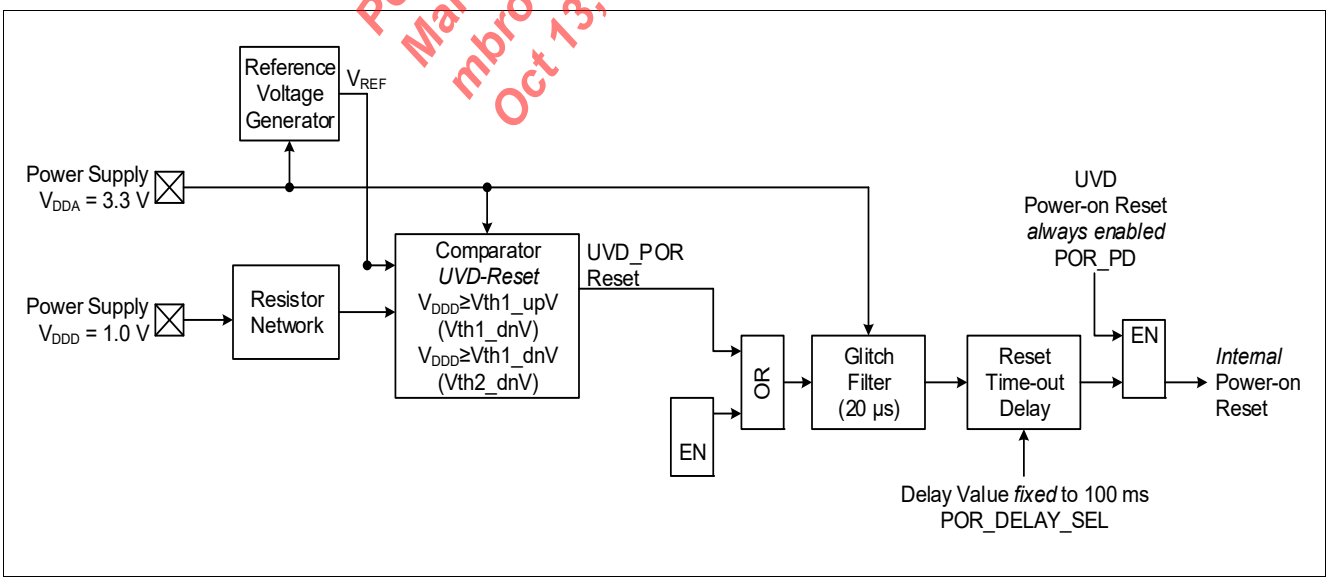


Figure 15 Reset System

4.4.2.2 Reset Control Unit (RCU)

4.4.2.2.1 Overview

The Reset Control Unit (RCU) provides these functionalities:

- A general hardware reset
- A hardware reset output signal connected to the $\overline{\text{HRST}}$ pin
- Individual software reset requests per peripheral of the system-on-chip

The main difference between general hardware reset and the software reset request is, that the general hardware reset affects all peripherals within the system-on-chip with the only exception of the PLLs. The software reset request can select dedicated peripheral(s) to be driven into reset, except the PLLs.

Table 70 TOP NoC Reset Source and Target Mapping

Module	POR/HW_RST	Global Software Reset	WDT	Software Reset Bit
BaseIA	R	R	R	R
NoC_Base	R	R	R	R
QSPI	R	R	R	R
Voice	R	R	R	R
ARC_SEM	R	R	R	R
Crypto	R	R	R	R
SPE	R	R	R	R
TOE	R	R	R	R
DMA3	R	R	R	R
SDXC[1)]	R	R	R	R
EMMC	R	R	R	R
HRST	R	R	R	R
SSB>NoC_Base	R	R	R	R
OTP>NoC_Base	R	R	R	R
FRW1>NoC_Base	R	R	R	R
FRW2>NoC_Base	R	R	R	R
DMA0	R	R	R	R
DDR_CTL0	R	R	R	R
DDR_CTL_APB	R	R	R	R

1) Not available in MxL25641

Table 71 Analog Module Reset Source and Target Mapping

Module	POR/HW_RST	Global Software Reset	WDT	Software Reset Bit
PCIe_PHY10	R	R	R	R
PCIe_PHY11	R	R	R	R
PCIe_PHY20	R	R	R	R
PCIe_PHY21	R	R	R	R
PCIe_PHY30[1)]	R	R	R	R
PCIe_PHY31[1)]	R	R	R	R
PCIe_PHY40[1)]	R	R	R	R
PCIe_PHY41[1)]	R	R	R	R
XFI_ETH	R	R	R	R
XFI_PON	R	R	R	R
USB_PHY0[1)]	R	R	R	R
USB_PHY1	R	R	R	R
PVT_Sensor	R	R	R	R
DDR_PHY	R	R	R	R
PLL0cz	R	NR	NR	R
PLL0cm0	R	NR	NR	R
PLL0cm1[1)]	R	NR	NR	R
PLL0b	R	NR	NR	R
PLL2	R	NR	NR	R
PLL1	R	NR	NR	R
PLLPP	R	NR	NR	R
LJPLL3	R	NR	NR	R
LJPLL4	R	NR	NR	R
DPLL	R	NR	NR	R
XO	NR	NR	NR	NR
POR	NR	NR	NR	NR
NRST_REG	R	NR	NR	NR (no software reset-able register)

1) Not available in MxL25641

Table 72 Peripheral NoC Reset Source and Target Mapping

Module	POR/HW_RST	Global Software Reset	WDT	Software Reset Bit
DMA0	R	R	R	R
EBU	R	R	R	R
PCM	R	R	R	R
SSC0 -->NoC4_Base	R	R	R	R
SSC1	R	R	R	R
SSC2	R	R	R	R
SSC3	R	R	R	R
I2C1	R	R	R	R
I2C2	R	R	R	R
I2S0	R	R	R	R
I2S1	R	R	R	R
I2C3	R	R	R	R
GPT0	R	R	R	R
GPT1	R	R	R	R
GPT2	R	R	R	R
ASC0 -->NoC4_Base	R	R	R	R
ASC1	R	R	R	R
ASC2	R	R	R	R
LEDC-->NoC4_Base	R	R	R	R
GPIO-->NoC4_Base	R	R	R	R

Table 73 Always ON NoC Reset Source and Target Mapping

Module	POR/HW_RST	Global Software Reset	WDT	Software Reset Bit
PMU(EPU)-->40MRST	R	R	R	NR
I2C0 -->NoCAON_Base	R	R	R	R
RCU-->40MRST	R	R	R	NR
ASC3	R	R	R	R
PVT Controller	R	R	R	R
CGU-->40MRST	R	R	R	NR

Table 74 HSIO NoC2/NoC3 Reset Source and Target Mapping

Module	POR/HW_RST	Global Software Reset	WDT	Software Reset Bit
PCIe_CTRL10	R	R	R	R
PCIe_CTRL11	R	R	R	R
PCIe_CTRL20	R	R	R	R
PCIe_CTRL21	R	R	R	R
SATA0_CTRL[1)]	R	R	R	R
SATA1_CTRL[1)]	R	R	R	R
PCIe_CTRL30[1)]	R	R	R	R
PCIe_CTRL31[1)]	R	R	R	R
PCIe_CTRL40[1)]	R	R	R	R
PCIe_CTRL41[1)]	R	R	R	R
SATA2_CTRL[1)]	R	R	R	R
SATA3_CTRL[1)]	R	R	R	R
NOC2/3_Base	R	R	R	NR

1) Not available in MxL25641

Table 75 ETH NoC Reset Source and Target Mapping

Module	POR/HW_RST	Global Software Reset	WDT	Software Reset Bit
iDMAr1	R	R	R	R
iDMAt1	R	R	R	R
iDMAt2	R	R	R	R
iDMAt0	R	R	R	R
GSWIP-O	R	R	R	R
PON[1)]	R	R	R	R
tMACSEC	R	R	R	R
XPCS5(XFILAN)	R	R	R	R
XPCSWAN(ETH)	R	R	R	R
USB0_CTRL[2)]	R	R	R	R
USB1_CTRL	R	R	R	R

1) Not available in URX850
2) Not available in MxL25641

Advanced Datapath Super Reset Domain

It is possible to reset the Advanced Datapath (ADP) with a single reset signal.

4.5 Interrupt System

The interrupt system provides the capability to react on external events by the means of hardware interrupts towards the CPU as described in these sections:

- **Features** (**Section 4.5.1**)
- **Functional Description** (**Section 4.5.2**)

4.5.1 Features

These are the interrupt system features:

- Indirect interrupts for each peripheral
- 256 interrupt resource, software remap to different CPU
  0 to 32 and 239 to 255 are reserved for the Intel Atom subsystem usage.
- Each GPIO can generate an external interrupt.
  Each external interrupt pin is individually programmable for edge or level detection.
- Software and hardware interrupt exception sources
- Up to 32 message-signaled interrupts for PCI Express EP
- Message-signaled interrupt for high performance peripherals
- No external NMI support

4.5.2 Functional Description

Table 76 describes the interrupt system.

4.5.2.1 Multiple Interrupt Path to the Intel Atom Subsystem

These are possible interrupt paths for certain peripherals:

- Signal connected to IOAPIC
- MSI message write to the Intel Atom subsystem via the central MSI generator (as per **Table 76**)
- MSI message write from external EP directly write into the Intel Atom subsystem

Table 76 Interrupt Signals to MSI_GEN

Module	MSI_GEN Pin	No. of Signals	Remarks
TSO, port_int[3:0]	msi_inp_intr_i[3:0]	4	Per port interrupt
LRO, m[31:00]_int_o	msi_inp_intr_i[35:4]	32	Eight pins to IOAPIC
MCPY, mcpy_int_o[7:0]	msi_inp_intr_i[43:36]	8	Per port interrupt
USB0[1)]	msi_inp_intr_i[44]	1	–
USB1	msi_inp_intr_i[45]	1	–
SATA, hsiol_stat[0]_intrq[1)]	msi_inp_intr_i[46]	1	HSIO L SATA 1 is no longer connected to MSI_GEN, it goes to IOAPIC only.
SATA, hsior_stat[0]_intrq[1)]	msi_inp_intr_i[48]	2	HSIO R SATA1 is no longer connected to MSI_GEN, it goes to IOAPIC only.
Crypto, ring_irq[3:0]	msi_inp_intr_i[53:50]	4	–
CQM, cqm_int_o[7:0]	msi_inp_intr_i[61:54]	8	–
GPTC2.timer 1	msi_inp_intr_i[47]	1	For potential NMI delivery
GPTC1.timer 1	msi_inp_intr_i[49]	1	For potential NMI delivery
GPTC0.timer 3	msi_inp_intr_i[62]	1	For potential NMI delivery
GPTC0.timer 2	msi_inp_intr_i[63]	1	For potential NMI delivery

1) Not available in MxL25641

MSI Generator

Figure 16 shows the MSI message format generated by a MSI generator uses the Intel Atom MSI definition.

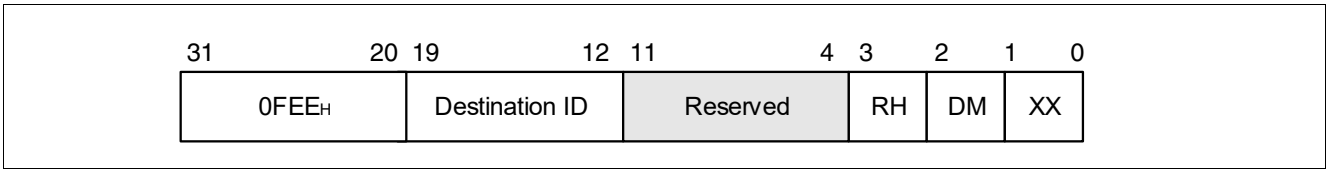


Figure 16 Intel Atom MSI Address Format

These are the fields in the message address register:

1. **Bits 31 - 20**
   These bits contain a fixed value for interrupt messages ($0FEE_H$). This value locates interrupts at the 1 Mbyte area with a base address of 4G - 18M. All accesses to this region are directed as interrupt messages.
   Note: Ensure that no other device claims the region as I/O space.
2. **Destination ID**
   This field contains an 8-bit destination ID. It identifies the message target processor(s). The destination ID corresponds to bits 63:56 of the I/O APIC redirection table entry when the IOAPIC is used to dispatch the interrupt to the processor(s).
3. **Redirection Hint indication (RH)**
   This bit indicates whether the message must be directed to the processor with the lowest interrupt priority among processors that can receive the interrupt.
   a) When RH is 0, the interrupt is directed to the processor listed in the destination ID field.
   b) When RH is 1 and the physical destination mode is used, the destination ID field must not be set to `0xFF`. It must point to a processor that is present and enabled to receive the interrupt.
   c) When RH is 1 and the logical destination mode is active in a system using a flat addressing model, the destination ID field must be set so that bits set to 1 identify processors that are present and enabled to receive the interrupt.
   d) When RH is set to 1 and the logical destination mode is active in a system using cluster addressing model, the destination ID field must not be set to `0xFF`. The processors identified with this field must be present and enabled to receive the interrupt.
4. **Destination Mode (DM)**
   This bit indicates whether the destination ID field must be interpreted as logical or physical APIC ID for delivery of the lowest priority interrupt.
   a) When RH is 1 and DM is 0, the destination ID field is in physical destination mode and only the processor in the system that has the matching APIC ID is considered for delivery of that interrupt. This means no re-direction.
   b) When RH is 1 and DM is 1, the destination ID field is interpreted as in logical destination mode and the redirection is only limited to processors that are part of the logical group of processors based on the processor's logical APIC ID and the destination ID field in the message.
   The logical group of processors consists of those identified by matching the 8-bit destination ID with the logical destination identified by the destination format register and the logical destination register in each local APIC.
   c) When RH is 0, the DM bit is ignored and the message is sent ahead independent of whether the physical or logical destination mode is used.

Figure 17 shows the layout of the message data register.

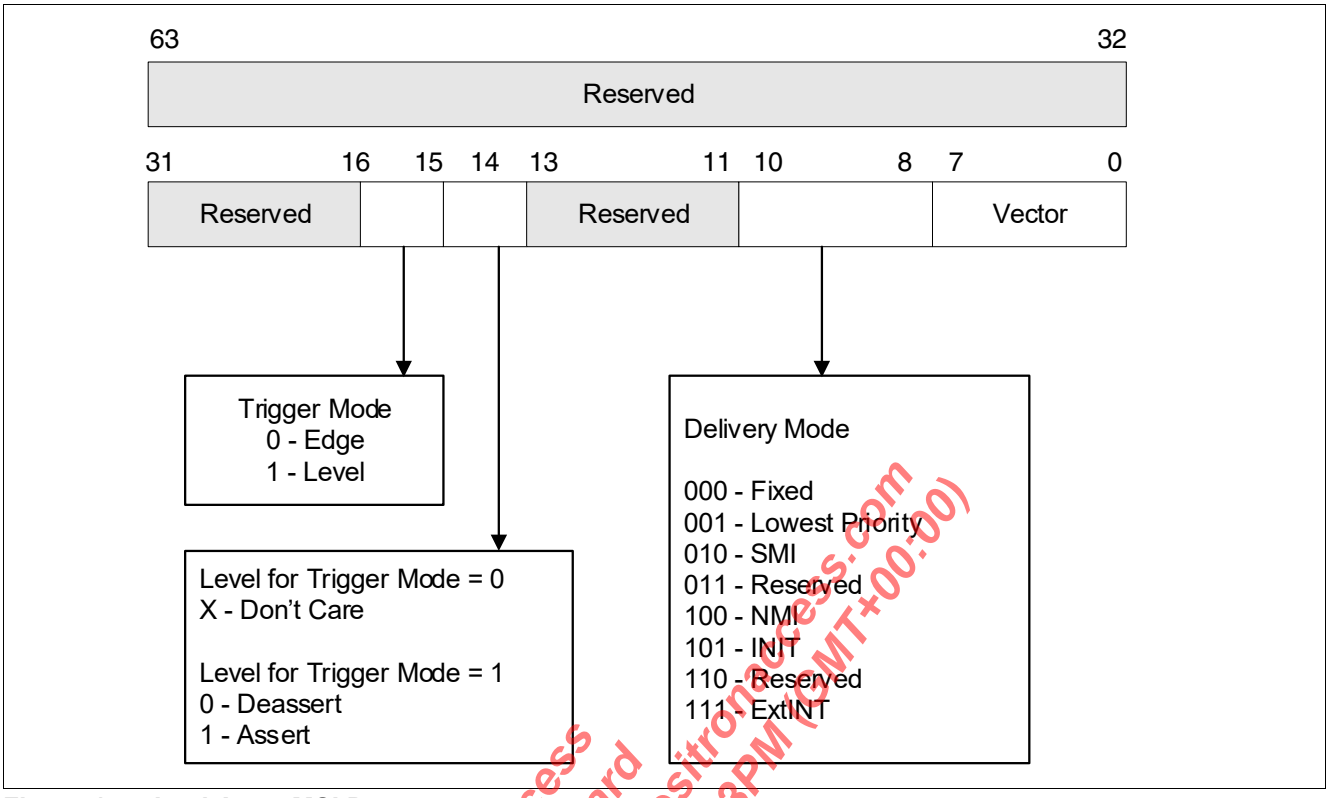


Figure 17 Intel Atom MSI Data

Reserved fields are assumed to be of no value. The software must preserve their contents on writes. These are other fields in the message data register:

1. **Vector**
   This 8-bit field contains the interrupt vector associated with the messages. Values range from 010_H to $0FE_H$. The software must guarantee that the field is not programmed with vector 00_H to $0F_H$.
2. **Delivery Mode**
   This 3-bit field specifies how the interrupt receipt is handled. The delivery modes operate only in conjunction with specified trigger modes. The correct trigger modes must be guaranteed by the software.
   These are restrictions:
   a) **000_B (Fixed Mode)**
      Delivers the signal to all the agents listed in the destination. The trigger mode for fixed delivery mode is either edge or level.
   b) **001_B (Lowest Priority)**
      Delivers the signal to the agent that is executing at the lowest priority of all agents listed in the destination field. The trigger mode is either edge or level.
   c) **010_B (System Management Interrupt or SMI)**
      The delivery mode is edge only. For systems that rely on SMI semantics, the vector field is ignored but must be programmed to all zeroes for future compatibility.
   d) **100_B (NMI)**
      Delivers the signal to all the agents listed in the destination field. The vector information is ignored. NMI is an edge-triggered interrupt regardless of the trigger mode setting.
   e) **101_B (INIT)**
      Delivers the signal to all the agents listed in the destination field. The vector information is ignored. INIT is an edge-triggered interrupt regardless of the trigger mode setting.

f) **111_B (ExtINT)**
Delivers the signal to the INTR signal of all agents in the destination field, as an interrupt that originated from an 8259A compatible interrupt controller. The vector is supplied by the INTA cycle issued by the activation of the ExtINT, which is an edge-triggered interrupt.

3. **Level**
Edge-triggered interrupt messages are always interpreted as assert messages. For edge-triggered interrupts, this field is not used. For level-triggered interrupts, this bit reflects the state of the interrupt input.
4. **Trigger Mode**
This field indicates the signal type that triggers a message.
a) **0** indicates edge sensitive.
b) **1** indicates level sensitive.

Example PCIe EP Generates Native MSI Writes

According to the PCIe base spec, MSI is a feature that enables an EP to request service by writing a system-specified data value to a system-specified address, using a DWORD memory write transaction. The system software initializes the message address and message data as a vector during device configuration, allocating one or more vectors to each MSI-capable EP.

This is an example:

MSI msg write == address + data in one PCIe TLP.

address== 0x FEE0 0000 (format defined by **Figure 16** register)

data = 0x 0000 0010 0000 0000 (format defined by **Figure 17** register)

4.5.2.2 Connecting Interrupt to IOAPIC

Table 77 summarizes the connections of all Interrupt Modules (IM) to the interrupt request lines assigned with the CPU.

Note: The IOAPIC has 256 pins, however, the Intel Atom and subsystem has used/reserved some of the interrupt vectors, as such the SoC level can have up to 205 interrupt pins, due to Intel Atom interrupt vector mapping. The IOAPIC pin to the Intel Atom internal interrupt vector mapping is programmable.

Table 77 Interrupt Interconnect System - Sorted Based on IRL

Peripheral Interrupt Source		IOAPIC
Request Line	**Description**	**Number for IOAPIC I/F**
--	Reserved for Intel Atom subsystem internal usage	–
		–
		–
		–
		–
		–
		–
		–
		–
PVT_IR	PVT controller interrupt	26
PCIe_10_INTA	PCIe core 10 interrupt A /MSI0	27
PCIe_10_INTB	PCIe core 10 interrupt B/MSI1	28

Table 77 Interrupt Interconnect System - Sorted Based on IRL (cont'd)

Peripheral Interrupt Source		IOAPIC
Request Line	**Description**	**Number for IOAPIC I/F**
PCIe_10_INTC	PCIe core 10 interrupt C /MSI2	29
PCIe_10_INTD	PCIe core 10 interrupt D /MSI3	30
DMA_FCC0_INT	DMA FCC 0 INT	31
DMA_FCC1_INT	DMA FCC 1 INT	32
DMA_FCC2_INT	DMA FCC 2 INT	33
PLL_LL_IR	All PLL (except Intel Atom subsystem PLLs) lock lost combined interrupt	34
SPI0_IR	SPI 0 combined interrupt	35
QSPI_IR	QSPI interrupt	36
TOH_ERR1	Interrupt for temperature overheat	37
TOH_ERR2	Interrupt for temperature overheat-pre-warning	38
MCPY	Interrupt for MCPY module	39
EBU_IR	EBU Interrupt	40
I2C0_IR	I^2C combined interrupt	41
I2S0_IR	I2S0 (OR from irq_srq_O[8:0], ERR	42
SDIO_IR[1)]	SDIO	43
EMMC_IR	eMMC	44
PWM_IR	PWM fan controller interrupt	45
USB0_OC	USB overcurrent detection or PON dying gasp	46
USB0_IR[1)]	USB controller 0	47
USB1_OC	USB overcurrent detection	48
USB1_IR	USB controller 1	49
VOICE_IR0	C55 voice subsystem interrupt	50
PCM1 TX EOP	DMA FCC PCM1 Tx End of Packet (EOP) interrupt	51
PCM1 RX EOP	DMA FCC PCM1 Rx EOP interrupt	52
PCM2 TX EOP	DMA FCC PCM2 Tx EOP interrupt	53
PCM2 RX EOP	DMA FCC PCM2 Rx EOP interrupt	54
XPCS_10_IR0	XPCS10 interrupt request	55
XPCS_11_IR0	XPCS11 interrupt request	56
XPCS_20_IR0	XPCS20 interrupt request	57
XPCS_21_IR0	XPCS21 interrupt request	58
XPCS_4_IR0	XPCS4 interrupt request	59
XPCS_5_IR0	XPCS5 interrupt request	60
TypeC_TCA_int0[1)]	TCA interrupt from USB 0	61
TypeC_TCA_int1[1)]	TCA interrupt from USB 1	62
TEP_INT	ARCSEM interrupt request	63
NAND_OD_IR	NAND odd page interrupt request	64
NAND_EV_IR	NAND even page interrupt request	65

Table 77 Interrupt Interconnect System - Sorted Based on IRL (cont'd)

Peripheral Interrupt Source		IOAPIC
Request Line	**Description**	**Number for IOAPIC I/F**
GSWIP-O_INT	GSWIP-O switch subsystem interrupt request	66
PCIe_10_IR	PCIe core 10 interrupt request	67
PCM1_IR	PCM1 transmit and receive interrupt	68
PCM0_IR	PCM0 transmit and receive interrupt	69
PCM2_IR	PCM2 transmit and receive interrupt	70
SPI_2	SPI 2 interrupt	71
SPI_3	SPI 3 interrupt	72
I2C_1	I^2C 1 interrupt	73
I2C_2	I^2C 2 interrupt	74
I2C_3	I^2C 3 interrupt	75
PCIe_11_INTA	PCIe core 11 interrupt A/MSI4	76
PCIe_11_INTB	PCIe core 11 interrupt B/MSI5	77
PCIe_11_INTC	PCIe core 11 interrupt C/MSI6	78
PCIe_11_INTD	PCIe core 11 interrupt D/MSI7	79
DMA0t_IR	DMA0t all channel interrupt	80
TSO_IR	Interrupt for ToE TSO	81
DMA0_IR	DMA0 (peripheral DMA) all channel interrupt	82
DMA1t_IR	DMA1t all channel interrupt	83
DMA1r_IR	DMA1r all channel interrupt	84
DMA2t_IR	DMA2t all channel interrupt	85
SATA0_IR[1)]	Interrupt for SATA controller 0	86
SATA1_IR[1)]	Interrupt for SATA controller 1	87
DPLL_HOLD_IR	HOLD IR	88
DPLL_FLOCK_IR	FLOCK IR	89
DPLL_PHASE_LOCK_IR	PHASE LOCK IR	90
SV_ERR_IR	Supervision circuit error IR	91
SATA2_IR[1)]	Interrupt for SATA controller 2	92
SATA3_IR[1)]	Interrupt for SATA controller 3	93
BIA_AXI_ERR_INT	Intel Atom subsystem AXI ports combined interrupt	94
–	–	95
–	–	96
–	–	97
–	–	98
EPU_IR0	EPU_INT0	99
EPU_IR1	EPU_INT1	100
GSWP_CORE_IR	GSWIP-O core interrupt	101
MACSEC_IG	MACsec ingress interrupt	102

Table 77 Interrupt Interconnect System - Sorted Based on IRL (cont'd)

Peripheral Interrupt Source		IOAPIC
Request Line	**Description**	**Number for IOAPIC I/F**
MACSEC_EG	MACsec egress interrupt	103
E197 Data	E197 data path interrupt	104
E197 DES0	E197 descriptor ring 0 interrupt	105
E197 DES1	E197 descriptor ring 1 interrupt	106
E197 DES2	E197 descriptor ring 2 interrupt	107
E197 DES3	E197 descriptor ring 3 interrupt	108
PCM0 TX EOP	DMA FCC PCM0 Tx EOP interrupt	109
PCM0 RX EOP	DMA FCC PCM0 Rx EOP interrupt	110
ASC1_IR	ASC 1 combined interrupt	111
8 kHz	8 kHz interrupt sourced from PLL1	112
HPNAND_INT	High-performance NAND module Write_Complete interrupt	113
I2S1_IR	I2S (OR from irq_srq_O[8:0],ERR	114
SPI1_IR	SPI 1 combined interrupt	115
PCIe_20_INTA	PCIe core 20 interrupt A /MSI8	116
PCIe_20_INTB	PCIe core 20 interrupt B /MSI9	117
PCIe_20_INTC	PCIe core 20 interrupt C /MSI10	118
PCIe_20_INTD	PCIe core 20 interrupt D/MSI11	119
PCIe_21_INTA	PCIe core 21 interrupt A /MSI12	120
PCIe_21_INTB	PCIe core 21 interrupt B /MSI13	121
PCIe_21_INTC	PCIe core 21 interrupt C /MSI14	122
PCIe_21_INTD	PCIe core 21 interrupt D /MSI15	123
–	–	124
–	–	125
–	–	126
–	–	127
ASC0_IR	ASC 0 combined interrupt	128
ASC2_IR	ASC 2 combined interrupt	129
ASC3_IR	ASC 3 combined interrupt	130
VOCODEC_0_IR	Voice codec 0 interrupt	131
VOCODEC_1_IR	Voice codec 1 interrupt	132
–	–	133
–	–	134
GPIO_INT0IR	GPIO 0-31 indirect interrupt	135
GPIO_INT1IR	GPIO 32-63 indirect interrupt	136
GPIO_INT2IR	GPIO 64-95 indirect interrupt	137
GPIO_INT3IR	GPIO 96-104 indirect interrupt	138
PON_IP_RX_INT[2)]	PON IP Rx INT	139

Table 77 Interrupt Interconnect System - Sorted Based on IRL (cont'd)

Peripheral Interrupt Source		IOAPIC
Request Line	**Description**	**Number for IOAPIC I/F**
PON_IP_TX_INT[2)]	PON IP Tx INT	140
GPIO_INT4IR	GPIO AON pins ORed indirect interrupt	141
MC_IR	Memory controller and IED interrupt	142
PCIe_11_IR	PCIe core 11 interrupt request	143
PCIe_20_IR	PCIe core 20 interrupt request	144
PCIe_21_IR	PCIe core 21 interrupt request	145
PCIe_30_IR[1)]	PCIe core 30 interrupt request	146
PCIe_31_IR[1)]	PCIe core 31 interrupt request	147
PCIe_40_IR[1)]	PCIe core 40 interrupt request	148
PCIe_41_IR[1)]	PCIe core 41 interrupt request	149
PON_LOS_IR[2)]	PON serdes_los_ir interrupt	150
MCPY_Port0	MCPY port interrupt	151
MCPY_Port1	MCPY port interrupt	152
MCPY_Port2	MCPY port interrupt	153
MCPY_Port3	MCPY port interrupt	154
MCPY_Port4	MCPY port interrupt	155
MCPY_Port5	MCPY port interrupt	156
MCPY_Port6	MCPY port interrupt	157
MCPY_Port7	MCPY port interrupt	158
LRO_SC_INT	LRO stall context interrupt	159
	MPS status interrupt #0	160
	MPS status interrupt #1	161
	MPS status interrupt #2	162
	MPS status interrupt #3	163
	MPS status interrupt #4	164
	MPS status interrupt #5	165
	MPS status interrupt #6	166
	Semaphore interrupt to the Intel Atom	167
	Global interrupt to the Intel Atom	168
NOCERR_IR0	NoCs Error interrupt Request	169
PPv4_IRQ	PPv4 indirect interrupt. All events are captured inside PPv4.	170
PONIP_IR[2)]	PON IP	171
E130_0	SPE DMA interrupt 0 to the Intel Atom	172
E130_1	SPE DMA interrupt 1 to the Intel Atom	173
E130_2	SPE DMA interrupt 2 to the Intel Atom	174
E130_3	SPE DMA interrupt 3 to the Intel Atom/TEP	175
WDT_ARC_VPN	WDT interrupt from VPN ARC	176

Table 77 Interrupt Interconnect System - Sorted Based on IRL (cont'd)

Peripheral Interrupt Source		IOAPIC
Request Line	**Description**	**Number for IOAPIC I/F**
Res	System call/trap	177
–	–	178
–	–	179
–	–	180
–	–	186
TC0_1A	Timer 0 16-bit counter 1A interrupt	188
TC0_1B	Timer 0 16-bit counter 1B interrupt	189
TC0_2AB	Timer 0 32-bit counter 2AB interrupt (Intel Atom Core 0 pre-warning)	190
TC0_3AB	Timer 0 32-bit counter 3AB interrupt (Intel Atom Core 1 pre-warning)	185
TC1_1AB	Timer 1 32-bit counter 1AB interrupt (Intel Atom Core 2 pre-warning)	187
TC1_2AB	Timer 1 32-bit counter 2AB interrupt (Intel Atom core 0 WDT)	–
TC1_3AB	Timer 1 32-bit counter 3AB interrupt (Intel Atom core 1 WDT)	–
TC2_1AB	Timer 2 32-bit counter 1AB interrupt (Intel Atom Core 3 pre-warning)	184
TC2_2AB	Timer 2 32-bit counter 2AB interrupt (Intel Atom core 2 WDT)	–
TC2_3AB	Timer 2 32-bit counter 3AB interrupt (Intel Atom core 3WDT)	–
TC3_1AB	Timer 3 32-bit counter 1AB interrupt	183
TC3_2AB	Timer 3 32-bit counter 2AB interrupt	181
TC3_3AB	Timer 3 32-bit counter 3AB interrupt	182
FRW_INT0IR	Firewall peripherals interrupt (ORed all firewalls interrupt other than the DDR, SSB firewall)	191
FRW_INT0IR	Firewall 0 DDR interrupt	192
FRW_INT1IR	Firewall 1 SSB interrupt	193
VM_INTIR	VM security violation error interrupt	194
DDR_CA_ERR_INTIR	DDR4 CA error interrupt	195
CQM_INT0	CQM interrupt 0	196
CQM_INT1	CQM interrupt 1	197
CQM_INT2	CQM interrupt 2	198
CQM_INT3	CQM interrupt 3	199
CQM_INT4	CQM interrupt 4	200
CQM_INT5	CQM interrupt 5	201
CQM_INT6	CQM interrupt 6	202

Table 77 Interrupt Interconnect System - Sorted Based on IRL (cont'd)

Peripheral Interrupt Source		IOAPIC
Request Line	**Description**	**Number for IOAPIC I/F**
CQM_INT7	CQM interrupt 7	203
CQM_INT8	CQM interrupt 8 (unused)	204
CQM_INT9	CQM interrupt 9 (unused)	205
LRO_Mod_IR0	LRO module interrupt 0	206
LRO_Mod_IR1	LRO module interrupt 1	207
LRO_Mod_IR2	LRO module interrupt 2	208
LRO_Mod_IR3	LRO module interrupt 3	209
LRO_Mod_IR4	LRO module interrupt 4	210
LRO_Mod_IR5	LRO module interrupt 5	211
LRO_Mod_IR6	LRO module interrupt 6	212
LRO_Mod_IR7	LRO module interrupt 7	213
LRO_ModER_IR	LRO module hardware error interrupt	214
LRO_EXP_IR	LRO exception interrupt	215
PCIe_30_INTA[1)]	PCIe core 30 interrupt A /MSI16	216
PCIe_30_INTB[1)]	PCIe core 30 interrupt B /MSI 17	217
PCIe_30_INTC[1)]	PCIe core 30 interrupt C /MSI18	218
PCIe_30_INTD	PCIe core 30 interrupt D /MSI19	219
PCIe_31_INTA[1)]	PCIe core 31 interrupt A /MSI20	220
PCIe_31_INTB[1)]	PCIe core 31 interrupt B /MSI21	221
PCIe_31_INTC[1)]	PCIe core 31 interrupt C /MSI22	222
PCIe_31_INTD[1)]	PCIe core 31 interrupt D /MSI23	223
PCIe_40_INTA[1)]	PCIe core 40 interrupt A /MSI24	224
PCIe_40_INTB[1)]	PCIe core 40 interrupt B /MSI25	225
PCIe_40_INTC[1)]	PCIe core 40 interrupt C /MSI26	226
PCIe_40_INTD[1)]	PCIe core 40 interrupt D /MSI27	227
PCIe_41_INTA[1)]	PCIe core 41 interrupt A /MSI28	228
PCIe_41_INTB[1)]	PCIe core 41 interrupt B /MSI29	229
PCIe_41_INTC[1)]	PCIe core 41 interrupt C /MSI30	230
PCIe_41_INTD[1)]	PCIe core 41 interrupt D/MSI31	231
–	Not for SoC usage	232 to 255

1) Not available in MxL25641
2) Not available in URX850

4.6 Boot System

The boot system describes the boot behavior, as described in these sections:

- **Features** (**Section 4.6.1**)
- **Functional Description** (**Section 4.6.2**)

The ROM code is at Intel Atom subsystem boot vector `0xFFFF FFF0`.

4.6.1 Features

The boot system offers these features:

- Boot from internal ROM using UART RS232 (ASC0).
- Boot from ROM and jump to 8-bit NAND using software sequencer
  - Boot from generic NAND flash (including different NAND parameter info in the BootRoM header)
  - Boot from Samsung NAND using NAND parameter info (in the BootRoM header) to boot
  - Boot from ONFI-compliant NAND using NAND parameter info (in the BootRoM header) to boot
- ECC support in 8/16-bit NAND hardware using ROM to configure the mode based on BRM encodings
- Up to 4 duplicated images for booting the system reliably
- System register programming
  - Register programming via encapsulated command from flash
  - White-list register access to prevent wrongly writing into unintended register
  - Range 2: `0xe000 0000` to `0xe07F FFFF`, 8 MB for CGU, RCU
  - GPTC Range: `0xE2500000H` to `0xE256FFFFH`
- Boot modes
  - Root of trust secure boot for production provisioning
  - Secure boot for secure boot-enabled end products
  - EEPROM modes are not supported in secure boot.
  - The default is non-secure boot.
- 160 Kbyte ROM
- Booting from QSPI interface with internal ROM
  - Boot from external serial flash
  - Boot from external serial EEPROM
  - 1 bit mode of SPI NOR, SPI NAND
  - 4 bit mode of SPI NOR, SPI NAND
- Boot from eMMC NAND with internal ROM using eMMC controller
- Boot from USB DFU mode
- Boot from SPI NAND with internal ROM using QSPI supporting:
  - Boot from external SPI SLC NAND flash
  - Boot from Micron/GigaDevice SLC NAND flash
  - Boot from 1GB, 2GB, 4GB size SLC NAND flash
  - Boot from SPI SLC NAND flash with hardware ECC embedded inside device

4.6.2 Functional Description

This section describes the hardware conditions after POR and describes the boot loader for on-chip ROM implementation. This boot loader stores codes in hardware to:

1. Pre-configure the system.
2. Handle the (boot) code download from an external non-volatile memory through a target peripheral interface.

4.6.2.1 Boot Selection after POR

The hardware status after POR is as follows:

- All Intel Atom subsystem hardware module interrupts are disabled.
- A reset status register in RCU stores the on-board pin strapping information.
- Boot mode strapping setting can be overridden by OTP bits.
- UART (MaxLinear and X-modem), DFU cannot be disabled by the overridden bit.

This boot selection process is done by hardware, there is no ROM code involvement yet.

Virtual Address Mapping

For software, the boot address is always fixed and started at (Intel Atom-specific) FFFF FFF0$_H$. This is a 32-bit real physical address.

Table 78 lists the pin strapping information determining the execution of boot code.

Table 78 Boot Mode Settings [1)]

Strap Select						
BOOT 3 (IO14_URT2_RTS_SPI1CS0_VRST1)	BOOT 2 (IO19_URT2_CTS_SPI1CK_SSI1CK)	BOOT 1 (IO03_CKO2_I2S1CLK)	BOOT 0 (IO67_UART1TX_TDM2DO)	HEX	First Boot Source	Second Boot Source
0	0	0	0	0x00	ROM	**QSPI NAND 1-bit (Fixed Mode)**
0	0	0	1	0x01		**USB Device Mode** DFU boot mode *Note: When other boot modes fail to load a bootable image, the ROM attempts to load an image via the USB1 interface. This is intended as system recovery. It is possible to permanently disable system recovery via feature credential.*
0	0	1	0	0x02		**UART0 X-Modem** This mode does not search for an connected EEPROM and directly fetch code from UART.
0	0	1	1	0x03		**QSPI NAND 1-bit (ONFI Mode)**
0	1	0	0	0x04		**UART0 MaxLinear Proprietary Mode** This mode uses the legacy MaxLinear mode for UART communication, only via UART0.
0	1	0	1	0x05		**QSPI NAND 4-bit (Fixed Mode)**
0	1	1	0	0x06		16-bit NAND golden block mode: boot from 16-bit NAND with golden block 0

Table 78 Boot Mode Settings (cont'd)[1)]

Strap Select						
BOOT 3 (IO14_URT2RTS_SPI1CS0_VRST1)	**BOOT 2 (IO19_URT2CTS_SPI1CK_SSI1CK)**	**BOOT 1 (IO03_CKO2_I2S1CLK)**	**BOOT 0 (IO67_URT1TX_TDM2DO)**	**HEX**	**First Boot Source**	**Second Boot Source**
0	1	1	1	0x07	Reserved	
1	0	0	0	0x08	ROM	**QSPI (EEPROM or Serial Flash)** CS = SPI_CS1,GPIO57 must be used for boot device, single bit mode, NOR.
1	0	0	1	0x09		**BE- NAND Boot** CS = NAND_CS1,GPIO 23 Features: • Device has embedded ECC. • Used for 1 KB/2 KB base NAND flash types • 5 address cycle • READ NAND parameters from flash embedded BRH • ROM does not perform ECC.
1	0	1	0	0x0A		**Generic (non-ONFI) NAND Boot** CS = NAND_CS1 Features: • Uses software BCH ECC. • Used for 1 KB/2 KB base NAND flash types • 5 address cycle • READ NAND parameters from flash embedded BRH
1	0	1	1	0x0B		**QSPI (Serial Flash)** CS = SPI_CS1 must be used for boot device, Quad SPI mode, NOR.
1	1	0	0	0x0C		**QSPI (Serial Flash)** CS = SPI_CS1 must be used for boot device, Quad SPI mode, NAND. (Golden Blk 0)
1	1	0	1	0x0D		**QSPI (Serial Flash)** CS = SPI_CS1 must be used for boot device, single bit SPI mode, NAND. (Golden Blk 0)

Table 78 Boot Mode Settings (cont'd)[1)]

Strap Select						
BOOT 3 (IO14_URT2RTS_SPI1CS0_VRST1)	**BOOT 2 (IO19_URT2CTS_SPI1CK_SSI1CK)**	**BOOT 1 (IO03_CKO2_I2S1CLK)**	**BOOT 0 (IO67_URT1TX_TDM2DO)**	**HEX**	**First Boot Source**	**Second Boot Source**
1	1	1	0	0x0E	ROM	**ONFI compliant NAND** CS = NAND_CS1 Features: • Uses software BCH ECC. • Used for 1 KB/2 KB Base NAND flash types • 5 address cycle • READ NAND parameters from flash embedded BRH and ONFI parameter page
1	1	1	1	0x0F		**eMMC NAND** Features: • Uses eMMC device based ECC.

1) There are three additional boot details from Boot ROM Header (BRH) for boot from HW BCH ECC to decode the page size, BCH algorithm, etc.

4.7 CPU Peripherals and Debugging

The CPU peripherals and debugging description covers these sections:

- **General Purpose Timer Counter** (**Section 4.7.1**)
- **Trace and JTAG Debug System** (**Section 4.7.2**)
- **UART Debugging** (**Section 4.7.3**)

4.7.1 General Purpose Timer Counter

There are four identical GPTC modules integrated.

- Instantiate one GPTC module in the AON NoC domain
  - Support fixed kernel clock of 40 MHz
  - Three GPTC 32-bit timer total
- Instantiate 3 GPTC modules in the peripherals NoC
  - Support fixed kernel clock of 200 MHz
  - Nine GPTC 32-bit timer total
  - Four GPTC 32-bit timers are used for the Intel Atom core WDT
  - Each Intel Atom core gets one pre-warning interrupt from one GPTC 32-bit timer
  - Each Intel Atom core gets one WDT reset base on one GPTC timer 32 bits

4.7.1.1 Features

Each GPTC provides these features:

- Six 16-bit timers
- Three single timer/counter blocks (timer1, 2, 3) consisting of two 16-bit timer/counter blocks (A, B)
- Operation as two separate timer/counter blocks in 16-bit mode or a single concatenated timer/counter in 32-bit mode
- Programmable counting direction
- Run/stop control allows to stop and restart the timer/counter by software at any time
- Software controlled reload allows to restart at the current value or at the reload value
- Auto reload function enables cyclic timer/counter operation
- Operation as timer or as counter
- Programmable edge detection of external count input or level sensitive counting
- Programmable inversion of the external input can be either:
  - Synchronized, to count asynchronous events at 1/2 of the GPTC clock frequency
  - Unsynchronized, for clock-synchronous events up to the GPTC clock frequency
- Interrupt generation at overflow

4.7.1.2 Overview

Figure 18 shows the overview and GPTC alternate input sources of 8 kHz clock.

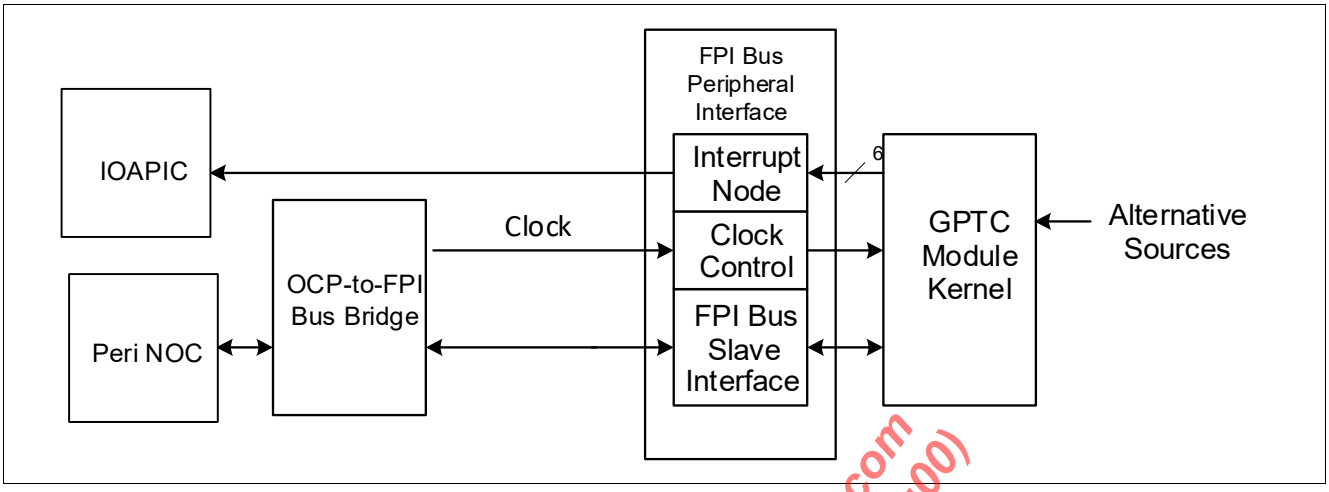


Figure 18 GPTC0/1/2 Alternate Input Source

4.7.2 Trace and JTAG Debug System

Powerful trace and debug capabilities are essential for systems which consists of multiple processing elements because the interaction between the processing elements is of major interest.

The URX85x features these processing elements:

- Multiple Intel Atom CPUs
- TEP
- Microcontroller inside the PPv4/PON subsystem
- DSP core inside the voice subsystem
- Complex interconnect instances

These sections describe the trace and debug capabilities of URX85x.

- Tracing is the capability to protocol the activity of processors/buses into a trace interface and can be started and stopped by different events.
- Debugging is done with the help of a debugger via the JTAG interface. In a debugging session, the processing elements can be halted by different break events.

4.7.2.1 Features

The multi core debug system provides these features:

- JTAG debug system
  - The debug host is connected via Ethernet or USB to the JTAG probe.
  - The JTAG probe is connected via a standard IEEE 1149.1-compliant JTAG interface to the JTAG controller.
  - The JTAG TAP controllers are daisy chained/individual opt out from the chain with CLTAP.
    IRR: DFx_CLTAPC_PIC, ID 99025184, RTL1P0_PIC5_V
- Test Control Unit (TCU) is implemented for hardware testing, for example boundary scan by the SoC not from CLTAP.
- Intel Atom trace interface and trace buffer off chip via 16-bit parallel MIPI interface. Up to 100 MHz per pin.
- NoC trace interface via same MIPI interface as previously described
- Performance counters for CPU cluster
- Four UART modules are available at chip-top level to be assigned to any of the embedded processors.
- SPI as system slave interface for connecting to external SPI master, internally connected at the NoC as bus master to access the full SoC resources.

4.7.2.2 Intel Atom JTAG Debugging

Most of the debug facilities are provided by the Intel Atom cores and the system agents inside the Intel Atom subsystem. These debug facilities are used and extended by additional hardware to support the debugging of other processing elements and interfaces.

The Intel Atom hardware is configurable to a certain extent, the main RTIT trace and debug functionality is determined by the Intel Atom debugging concept.

The Intel Atom processor provides these debugging facilities:

- Debug exception (#DB)
  Transfers program control to a debug procedure or task when a debug event occurs.
- Breakpoint exception (#BP)
- Breakpoint-address registers (DR0 through DR3)
  Specifies the addresses of up to four breakpoints.
- Debug status register (DR6)
  Reports the conditions that were in effect when a debug or breakpoint exception was generated.
- Debug control register (DR7)
  Specifies the forms of memory or I/O access that cause breakpoints to be generated.
- T (trap) flag, TSS, RF (resume) flag, EFLAGS register, TF (trap) flag, EFLAGS register
- Breakpoint instruction (INT 3)
- Last branch recording facilities

JTAG is the primary debug interface for the Intel Atom cores. However, the debugger access to this JTAG interface is protected at two levels.

Enabling the JTAG interface requires successful authentication via a challenge response mechanism, which can be done back to a server in the cloud.

The second level provides different access levels.

4.7.2.3 Intel Atom RTIT Trace

The trace facility allows to trace the program flow of a processing element. The main object of trace is to show the exact program flow from a specific program execution or just a small window of the execution, in parallel to primary processor pipeline. The RTIT trace data is sent out at the PTI parallel interface for external debugger to decode and analyze.

4.7.2.3.1 Overview

The Intel Atom subsystem provides a 256 bytes trace information from each Intel Atom core.

The software master ID 128 is reserved for RTIT tracing.

The RTIT base address is:

RTIT core 0 (masterID 128, ch 0, block 0) = 0xf4c0_0000

RTIT core 1 (masterID 128, ch 1, block 0) = 0xf4c0_0100

RTIT core 2 (masterID 128, ch 2, block 0) = 0xf4c0_0200

RTIT core 3 (masterID 128, ch 3, block 0) = 0xf4c0_0300

4.7.2.4 Boundary Scan

The Boundary Scan Test (BST) is a standardized method for testing boards, providing also a standard interface to communicate with test circuits on ICs. The boundary scan standard specifies a four wire interface using the four signals TDI, TDO, TCK, and TMS. Additionally, it is possible to integrate an optional test reset signal $\overline{\text{TRST}}$. These five dedicated signals, the Test Access Port (TAP), are connected to the TAP controller. The TAP controller is a state machine clocked with the rising edge of TCK and the state transitions are controlled by TMS. The Test Control Unit (TCU) contains a TAP controller used to perform the boundary scan test. The TCU shares its pins with the JTAG debug system as shown in **Section 4.7.2.5**.

A fully IEEE 1149.1 compliant boundary scan support is provided with these functionalities:

- Complete boundary scan (excluded Ethernet PHY)
- A test access port controller (TAP controller) included in the TCU
- Five dedicated pins to asynchronously reset the TAP controller of the TCU independently from the internal logic and chip reset
  - TCK
  - TMS
  - TDI
  - TDO
  - $\overline{\text{TRST}}$
- One 32-bit IDCODE register
- Boundary Scan Description Language (BSDL)

BSDL is a standard way to describe the features and behavior of an IC, that includes IEEE 1149.1 boundary scan. It is also a standard way to pass information to test-generation software. Using BSDL, the test software also checks that the BST features are correct.

These instructions are supported for boundary scan:

- Bypass (Instruction Code = 11111111)
- Sample/Preload (Instruction Code = 00000010)
- Extest (Instruction Code = 00000000)
- IDCODE (Instruction Code = 00000100)
- Clamp (Instruction Code = 00000110)
- HighZ (Instruction Code = 00000111)
- The ID code value read out for different product

JTAG ID Readback Value

The JTAG read back device IDs are:

- 0x30090113 for URX851
- 0x30083113 for URX850
- 0x300A0113 for MxL25641

4.7.2.5 BS JTAG/ Debug JTAG Interface

Pin strapping in the TDO line during TRST defines whether the chip goes into BS test mode or chip debug mode. The JTAG for chip debug modes are daisy chained. A disabled module can be excluded from the daisy chain. **Chapter 11** describes the security settings required for secure CPU JTAG.

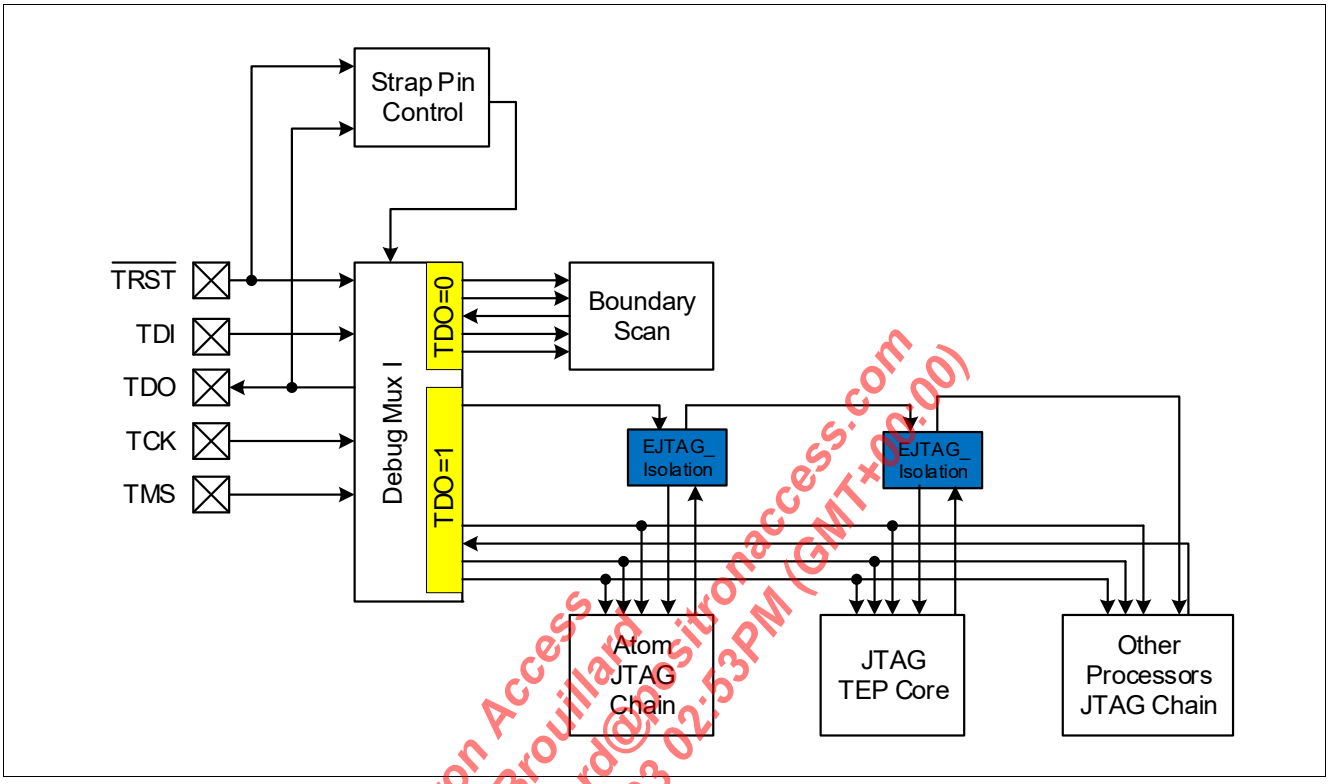


Figure 19 BS JTAG and Debug JTAG Pin Mux Detail

Attention: The strapping decision was swapped compared to the previous CABU products.
It was TDO=1 goes for BSCAN mode, now it is TDO=0 goes for BSCAN mode.
It was TDO=0 for system debug, now it is TDO=1 for system debug.

4.7.3 UART Debugging

There are four sets of UART interfaces, and three of them are muxed for debugging different processor elements. Via the muxing selection, it is possible to debug four Intel Atom cores at the same time or flexibly select an Intel Atom core and another processor to debug concurrently.

UART0 Muxing for Debugging

UART0 is muxed with ASC0 (Intel Atom-controlled UART), TEP UART, and VCODEC0 UART from SSI0, PON UART1.

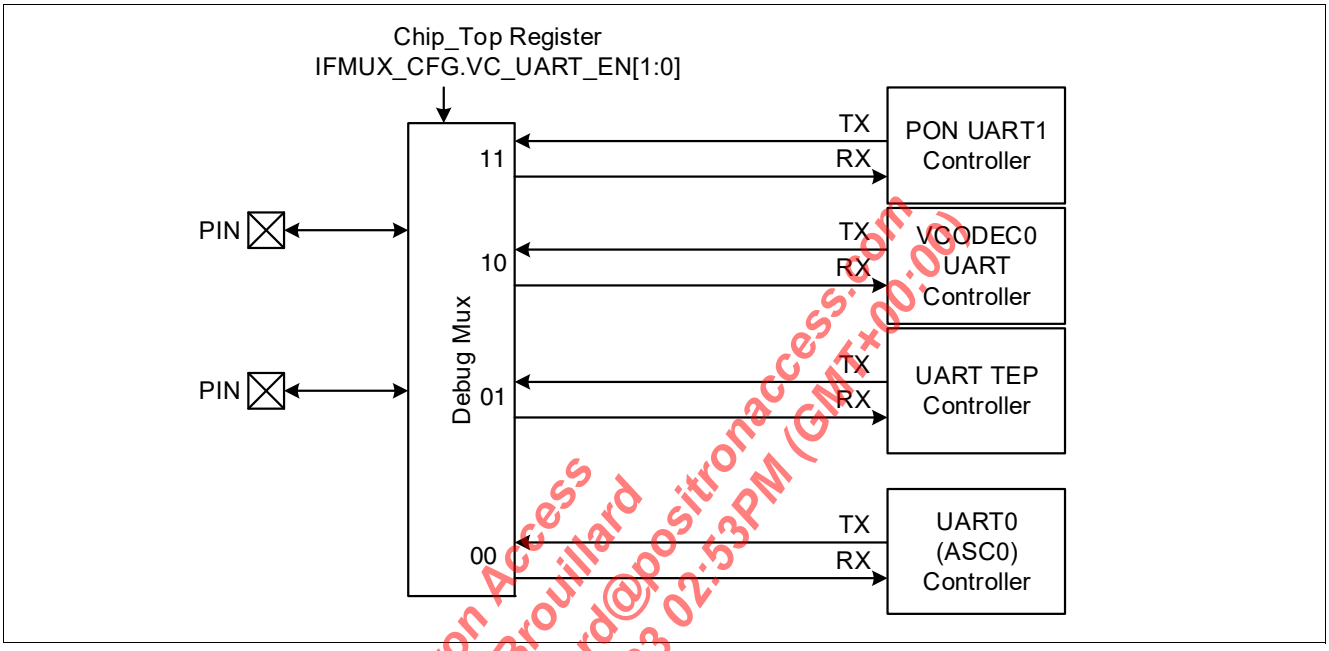


Figure 20 V-CODEC UART Port 0 for Debugging

Attention: Ball (Pin) Function Description provides the pin definition.

Attention: PON_UART1 is not available in URX850.

UART1 and 2 are muxed with ASC1/2 (Intel Atom-controlled UART), PON IP UART0 and C55 DSP UART and QoS microcontroller UARTs, VCODEC1 UART of SSI1, VPN UART.

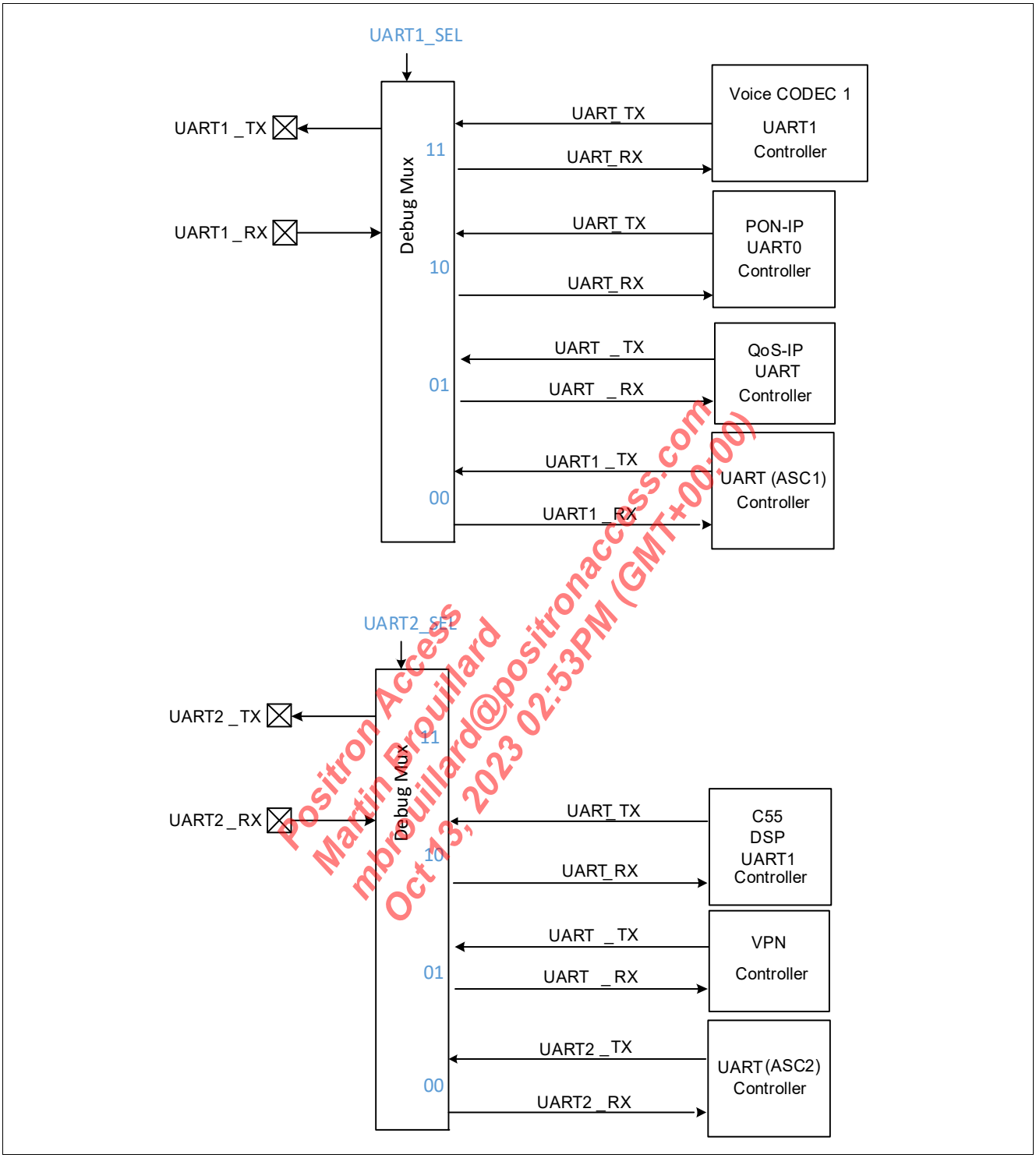


Figure 21 UART 1 and UART 2 Port for Debugging

Attention: PON-IP UART0 is not available in URX850.

5 Chip ID Register Description

Absolute Register Address = Module Base Address + Offset Address

Table 79 Registers Address Space

Module	Base Address	End Address	Note
CHIPID	$ECC80000_H$	$ECC800BE_H$	Chip ID Register Description

Table 80 Registers Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
MPS_MANID	Manufacturer Identification Register	00_H	00001120_H
MPS_CHIPID	Chip Identification Register	04_H	00080113_H

The register is addressed bytewise.

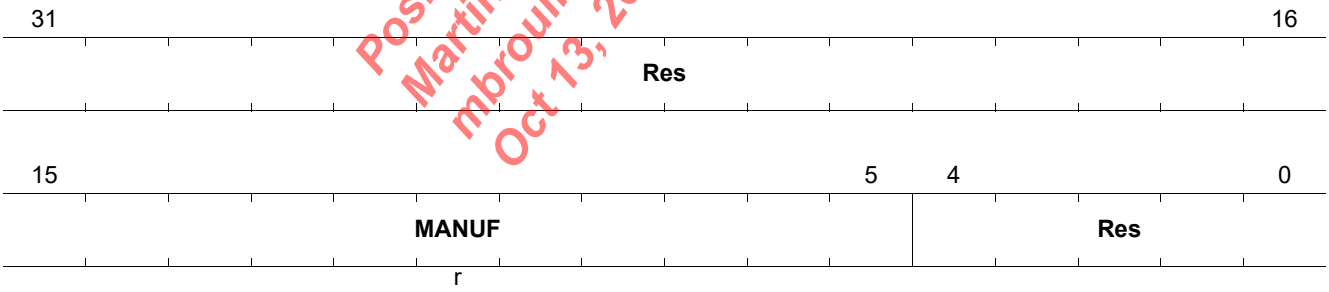
5.1 Chip ID Register Description

This section describes all registers of CHIPID in detail.

Manufacturer Identification Register

MPS_MANID **Offset** **Reset Value**

Manufacturer Identification Register **00_H** **00001120_H**

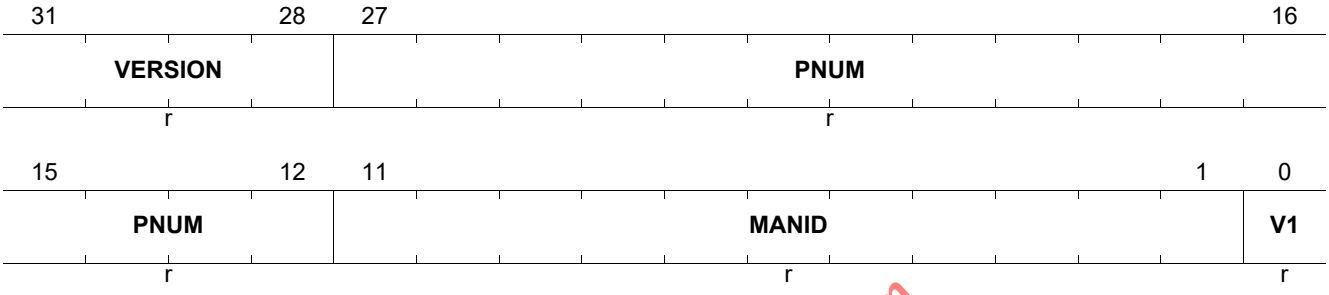


Field	Bits	Type	Description
MANUF	15:5	r	**Manufacturer Identification Number (JEDEC)** The JEDEC normalized manufacturer ID code for MaxLinear

Chip Identification Register

Identifier of the chip. The chip identification register is identical to the JTAG identification register.

MPS_CHIPID	Offset	Reset Value
Chip Identification Register	04_H	30090113_H



Field	Bits	Type	Description
VERSION	31:28	r	**Chip Version Number** The 4 bits are configured by fusing. A2: 1 A3: 2 B0: 3
PNUM	27:12	r	**Part Number** Coding is used to differentiate different product variants. 0000 0000 1000 0000 to 0000 0000 1011 1111. 64 variant. • URX850 = 0x83 • URX851 = 0x90 • MxL25641 = 0xA0
MANID	11:1	r	**Manufacturer ID** The JEDEC normalized manufacturer ID code for MaxLinear 0001 0001 0011
V1	0	r	**Fixed to 1** Bit is fixed to 1.

6 Memory Interfaces

The memory interfaces description covers these sections:

- **DDR3, DDR4 and LPDDR4 SDRAM Controller** (**Section 6.1**)
- **eMMC Interface Controller** (**Section 6.2**)
- **External Bus Unit NAND Controller** (**Section 6.3**)
- **Quad Serial Peripheral Interfaces** (**Section 6.4**)

6.1 DDR3, DDR4 and LPDDR4 SDRAM Controller

This section describes the DDR subsystem.

- **DDR SDRAM System Overview** (**Section 6.1.1**)
- **DDR4, DDR3, and LPDDR SDRAM Controller** (**Section 6.1.2**)
- **DDR SDRAM System Integration** (**Section 6.1.3**)

6.1.1 DDR SDRAM System Overview

The DDR-SDRAM system is connected directly to the NoC. This allows system devices direct access to the DDR memory via the NoC, such as the Intel Atom subsystem, PCIe devices, or the packet processor.

The DDR controller is configured to support up to 32-bit+8-bit ECC Data interface. With flexible byte lane assignment, the DDR interface can be system-friendly to support 32-bit, 32-bit+8-bit ECC, and 16-bit only interfaces.

Attention: The 8-bit ECC supports hamming code based 1-bit error detection and correction and 2-bit error detection. The ECC feature is not available on MxL25641 devices.

6.1.1.1 DDR Module List

The DDR subsystem consists of these modules:

- Universal Protocol Controller
- PHY Utility Block (PUB)
- DDR4/DDR3/LPDDR4 SDRAM PHY
- SSTL PADs

Table 81 DDR Interface Width

Models	DDR4	DDR3L	LPDDR4
32-bit + 8-bit ECC[1)]	Yes	Yes	No
32-bit	Yes	Yes	Yes
16-bit	Yes	Yes	No

1) Only the URX851 and URX850 devices support ECC.
The MxL25641 device does not support ECC.

6.1.2 DDR4, DDR3, and LPDDR SDRAM Controller

The DDR controller supports these features:

- Support x8, x16 memories for a memory data path of 32/16-bits interface
- Two memory ranks, devices within a rank tie to a common chip select
- Up to eight memory physical banks
- DDR3/DDR4/LPDDR4 protocols
- Efficient DDR protocol with in-order column commands and out-of-order activate and pre-charge commands
- Down to 5 clock cycles command latency
- Automatic power-down and self-refresh entry and exit, driven from software and hardware are supported
- Programmable per rank memory ODT
- Programmable data training assists in training of the data eye of the memory channel
- Programmable page policy
- Optimizes intelligent precharge/activate timings
- Read reorder buffer on the port to holds the data corresponding to a DDR command
- APB register interface at 32-bit data and 15-bit address width
  - The APB interface clock is synchronous to pclk
  - It must be same or less than the controller clock
- DDR controller core clock at half of the DDR interface clock (1:2 clock ratio)
- DDR4 features
  - Data Bus inversion (DBI)
  - Multi-Purpose Register (MPR) reads and writes
  - CA parity error without retry
  - Per DRAM Addressability (PDA), for example program ODT or Vref on a DRAM device of a given rank
  - Programmable/address latency
  - Write CRC
    The DRAM device compares the CRC checksum coming from the controller and flags CRC error.
- For LPDDR4, supports direct software request control or programmable internal control for ZQ calibration cycles
- For LPDDR4, supports for ZQ calibration after SR-power-down exit
- Support automatic SDRAM power-down entry and exit caused by lack of transaction arrival for a programmable time
- For URX851 and URX850, separate core voltage domain to support over drive the DDR controller to 933 MHz for LPDDR4-3733
- Support self-refresh entry and exit under different modes:
  - By a programmable time
  - Under full software control
  - Via DDRC hardware low power interface control

6.1.3 DDR SDRAM System Integration

6.1.3.1 Supported DDR3/4 SDRAM Speed Grades

Table 81 shows the speed grades of DDR3, DDR4, and LPDDR4 SDRAM devices supported by the DDR SDRAM controller.

Table 82 Supported DDR3/4/LPDDR4 SDRAM Speed Grades

Speed Grade	DDRx-667	DDRx-1600	DDRx-2133	DDRx-2667	DDR4-3200 LPDDR4-3200	LPDDR4-3733[1)]
SDRAM Device Clock	333 MHz	800 MHz	1066 MHz	1333 MHz	1600 MHz	1866 MHz

1) Over drive is required to achieve this frequency. Only for URX851 and URX850.

6.1.3.2 Supported DDR3/4 SDRAM Devices

Table 83 show the DDR3 and DDR4 SDRAM devices supported by the DDR SDRAM controller.

Table 83 Supported DDR3 and DDR4 SDRAM Devices

Depth	Width	No. of Banks	Row Addr. Width	Col. Addr. Width
512 Mbit DDR3 SDRAM Devices				
32M	x16	8	12	10
64M	x8	8	13	10
1 Gbit DDR3 SDRAM Devices				
64M	x16	8	13	10
128M	x8	8	14	10
2 Gbit DDR3 SDRAM Devices				
128M	x16	8	14	10
256M	x8	8	15	10
4 Gbit DDR3 SDRAM Devices				
256M	x16	8	15	10
512M	x8	8	16	10
8 Gbit DDR3 SDRAM Devices				
512M	x16	8	16	10
1G	x8	8	16	11
2 Gbit DDR4 SDRAM Devices				
128M	x16	4	14	10
256M	x8	8	14	10
4 Gbit DDR4 SDRAM Devices				
256M	x16	4	15	10
512M	x8	8	15	10
8 Gbit DDR4 SDRAM Devices				
512M	x16	4	16	10
1G	x8	8	16	10

Table 83 Supported DDR3 and DDR4 SDRAM Devices (cont'd)

Depth	Width	No. of Banks	Row Addr. Width	Col. Addr. Width
16 Gbit DDR4 SDRAM Devices[1)]				
1G	x16	4	17	10
2G	x8	8	17	10
32 Gbit DDR4 3DS SDRAM Devices[1)]				
4G	x8	16 per rank; 4 logic ranks	16	10

1) Only for URX851 and URX850

Table 84 show the supported LPDDR4 devices.

Table 84 Supported LPDDR4 Devices

Depth	Width	No. of Banks	Row Addr. Width	Col. Addr. Width
4 Gbit LPDDR4 SDRAM Devices				
256M	x16 x2 (channels)	8	14	10
8 Gbit LPDDR4 SDRAM Devices				
512M	x16 x2 (channels)	8	15	10
16 Gbit LPDDR4 SDRAM Devices[1)]				
1G	x16 x2 (channels)	8	16	10
32 Gbit LPDDR4 SDRAM Devices[1)]				
2G	x16 x2 (channels)	8	17	10

1) Only for URX851 and URX850

6.2 eMMC Interface Controller

This section describes the eMMC controller.

6.2.1 Features

The eMMC supports:

- Host clock rate variable between 0 to 200 MHz
- Up to 3200 Mbps data rate using 8 parallel data lines (eMMC HS400)
- Up to 1600 Mbps data rate using 8 parallel data lines (eMMC HS200)
- Up to 832 Mbps data rate using 8 parallel data lines (eMMC DDR52 mode)
- Data transfer in 1-bit, 4-bit, and 8-bit modes
- Cyclic redundancy check CRC7 for command and CRC16 for data integrity
- Support for eMMC plus and eMMC mobile
- Command queuing engine complies to the CQHCI spec (version 0.7)
- Integration of programmable PADs with driving strength/impedance, weak pull up/down control

6.3 External Bus Unit NAND Controller

The External Bus Unit (EBU) interfaces between external memories (flash device)/peripheral units and the internal system modules. The EBU operates as host only.

6.3.1 NAND Controller Subsystem Features

The NAND controller subsystem consisting of the EBU and ECC-NAND controller supports:

- NAND flash features
  - 16-bit/8-bit data bus
- Page + OOB bytes
  - 512+ (2 - 13) bytes per page
  - 2048+ (7 - 210) bytes per page
  - 4096+ (13 - 420) bytes per page
  - 8192+ (26 - 840) bytes per page
  - Supports 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, and 32768 Mbytes flash device
  - ECC calculation/generation and verification on-the-fly

6.3.2 Functional Description

The EBU is used for the communication with external memories or peripheral units.

6.3.2.1 NAND Access Enhancement Module

This section describes the enhancement of NAND access.

6.3.2.1.1 Features

The enhancements support:

- High speed NAND access with hardware based NAND access sequencer
- Hardware-based ECC calculation, checking, and correction

Support generic NAND Flash

Table 85 lists the intended NAND flash devices. However, it is possible for the user to select a device not present in the list after consideration of command compatibility and timing compatibility.

Table 85 Supported Generic NAND Devices

Vendor	Type	Package	Size	Page Size	Address Cycles
512 B Page Size					
S	SLC	TSOP-48	32M/64M x 8 bits	512 B+ 16 B	3: [7:0];[16:9];[24:17] 4: [7:0];[16:9];[24:17];[25]
N	SLC	TSOP-48	64M x 8 bits	512 B + 16 B	4: [7:0];[16:9];[24:17];[25]
2 kB Page Size					
Sa	SLC	TSOP-48	1G x 8 bits	2 kB+ 64 B	5: [7:0];[11:8];[19:12];[27:20];[30:28]
H	SLC	TSOP-48	16 Gbit	2 kB+64 B	5: [7:0];[11:8];[19:12];[27:20];[30:28]

Table 85 Supported Generic NAND Devices (cont'd)

Vendor	Type	Package	Size	Page Size	Address Cycles
M	SLC	TSOP-48	xF2: 2 Gbit xF4: 4 Gbit xF8: 8 Gbit	2 kB+64 B	5: [7:0] [11:8] [19:12] [27:20] xF2 = [28]; xF4/8 = [29:28]
4 kB Page Size					
N	MLC	TSOP48	64 Gbit	4 kB+128 B	
S	SLC	TSOP-48	8 Gbit	4 kB+128 B	5 Cycle: 1) [7:0] 2) [12:8] 3) [20:13] 4) [28:21] 5) [30:29]
M	MLC	TSOP-48	xF32: 32 Gbit xF64: 64 Gbit xF128: 128 Gbit xF256: 256 Gbit	4 kB+**224 B**	5 Cycle: 1) CA [7:0] 2) CA [12:8] 3) PA [7:0] 4) BA [15:8] 5) BA [19:16]
8 kB Page Size					
S	SLC	TSOP-48	8 Gbit	8 kB+**436 B**	5 Cycle: 1) CA [7:0] 2) CA [13:8] 3) PA [20:14] BA [21] 4) BA [29:22] 5) BA [31:30]
M		TSOP-48			

6.3.2.1.2 Functional Description

The supported modes are:

- NAND mode using EBU controller mode with embedded hamming code
  Due to backward compatibility, it is possible to use this mode only in combination with serial flash for BOOT. No boot is supported from the EBU.
- NAND mode using high speed NAND controller mode with embedded BCH ECC for on-the-fly ECC generation/validation/correction. **Boot Mode Settings** is supported via Boot ROM.

EBU NAND Mode (without High Speed NAND Controller)

With the NANDMODE signal, together with the EBU CSx active signal, the external NAND flash interface is selected for access. It means the EBU CSx range is overlapped with the NAND flash.

When the NAND flash interface is enabled, the CPU reads/writes the address lines to the NAND flash controller so that the NAND flash device can be accessed. Data is don't care except for cmd and data transfer.

NAND flash has both data and address multiplexed onto I/O lines. Command, address, and data are all written into the flash device via the I/Os during the low state (active) of WE and CE. The command latch and address latch are used to multiplex both the command and address respectively, via the same I/O pins.

The commands are written to the IO interface by a means of GPIO access the software delivers the READ and WRITE operations on the internal FPI address bus, and delivers commands on the internal FPI data bus in a write transfer.

Example: Reading Data from Address 0 in a 2 GB NAND Flash:

There are 5 address cycles for a 2 GB NAND flash (256 M X 8 bits).

The first cycle is column address bits [7:0], the second cycle is column address bits [11:8], the third cycle is row address bits [19:12], the fourth cycle is row address bits [27:20], and the fifth cycle is row address bit [28]. The second cycle delivers only A8-A11 and is aligned to IO bit 0, thus all bits on the IO bits [7:4] must be set to zeros. This applies to the fifth cycle.

1) In the read command latch cycle, the software writes data 00_H to the IO based memory address `0x1400_0018` (CS | CLE), then the address latch cycle follows the read command latch cycle.

2) The software writes data 00_H to the IO based memory address `0x1400_0014` (CS | ALE) for the first 8 bit address which carry column address bits [7:0],

3) 4) 5) The software repeats the same write transfer again and again for the second address cycle carrying 4 bit column address[11:8], third address cycle carrying 8 bit row address[19:12], and fourth address cycle carrying 8 bit row address[27:20].

6) To end the address cycle with the fifth address cycle carrying 1 bit row address[28], the software writes data 00_H to the IO based memory address `0x1400_0010` (CS_N).

High Speed NAND DMA Access Controller Mode

The high speed NAND DMA access controller supports READ, WRITE operations. Erase, Reset and READ ID must be done using the EBU NAND mode controller access.

Supported NAND Devices

Table 86 lists the intended NAND flash devices. However, it is possible for the user to select a device not present in the list after consideration of command compatibility and timing compatibility.

Table 86 Supported NAND Devices

Vendor	Type	Package	Size	Page Size	Address Cycles
512 B Page Size					
S	SLC	TSOP-48	32M/64M x 8 bits	512 B+ 16 B	3: [7:0];[16:9];[24:17] 4: [7:0];[16:9];[24:17];[25]
N	SLC	TSOP-48	64M x 8 bits	512 B + 16 B	4: [7:0];[16:9];[24:17];[25]
2 kB Page Size					
S	SLC	TSOP-48	1G x 8 bits	2 kB+ 64 B	5: [7:0];[11:8];[19:12];[27:20];[30:28]
H	SLC	TSOP-48	16 Gbit	2 kB+64 B	5: [7:0];[11:8];[19:12];[27:20];[30:28]
M	SLC	TSOP-48	xF2: 2 Gbit xF4: 4 Gbit xF8: 8 Gbit	2 kB+64 B	5: [7:0] [11:8] [19:12] [27:20] xF2 = [28]; xF4/8 = [29:28]
4 kB Page Size					
N	MLC	TSOP48	64 Gbit	4 kB+128 B	
S	SLC	TSOP-48	8 Gbit	4 kB+128 B	5 Cycle: 1) [7:0] 2) [12:8] 3) [20:13] 4) [28:21] 5) [30:29]
M	MLC	TSOP-48	xF32: 32 Gbit xF64: 64 Gbit xF128: 128 Gbit xF256: 256 Gbit	4 kB+**224 B**	5 Cycle: 1) CA [7:0] 2) CA [12:8] 3) PA [7:0] 4) BA [15:8] 5) BA [19:16]

Table 86 Supported NAND Devices (cont'd)

Vendor	Type	Package	Size	Page Size	Address Cycles
8 kB Page Size					
S	SLC	TSOP-48	8 Gbit	8 kB+**436 B**	5 Cycle: 1) CA [7:0] 2) CA [13:8] 3) PA [20:14] BA [21] 4) BA [29:22] 5) BA [31:30]
M	MLCx3	TSOP-48	64 Gbit	8 kB+976 B	5 Cycle: 1) CA [7:0] 2) CA [13:8] 3) PA [7:0] 4) PA [8], BA [15:9] 5) BA [20:16], LA [0]

NAND Controller Block Diagram

The hardware interface to the NAND device is kept the same as of today, connecting to the NAND device by multiplexed pins of EBU signals. The chip select to NAND device is either CS0 or CS1.

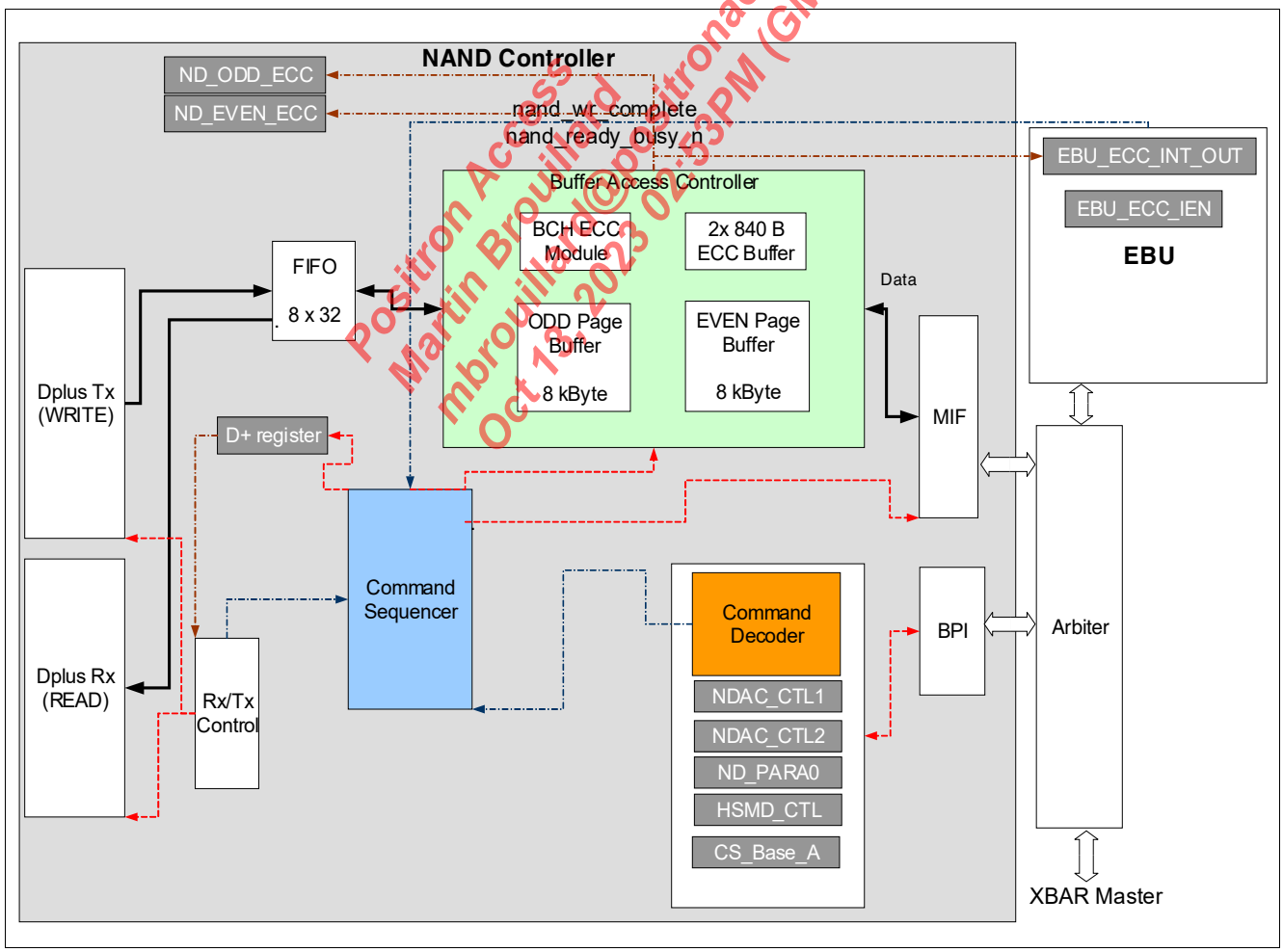


Figure 22 NAND Controller - Top Level Block Diagram

Command Sequencer

The sequencer is a collect of state machines which represents a specific NAND access. The sequencer has state machines to read and write data from/to the flash using the MIF interface. Once triggered by the scheduler for a read/write, the sequencer generates the command cycle (read/write), address cycles (3, 4, or 5) and the data cycles through the MIF interface. Once the data read/write is completed, the sequencer gives an operation done signal to the scheduler and returns to the idle state.

BCH ECC Module

This section describes the core error correction requirements/capabilities of the BCH ECC system. Integration details, such as hardware interfaces, locations in memory, and other related components are not addressed. Table 87 and Table 88 provide all supported ECC block sizes, page sizes, and correction capabilities.

Table 87 BCH ECC HW for 512 B Info Block Size

	Info Block Size = 512 B											
Page Size (Bytes)	512	512	512	2048	2048	2048	4096	4096	4096	8192	8192	8192
Bits Correctable per ECC CW	1	4	8	1	4	8	1	4	8	1	4	8
No. of ECC CWs per Page	1	1	1	4	4	4	8	8	8	16	16	16
log2 (Galois Field Order)	14	14	14	14	14	14	14	14	14	14	14	14
Maximum No. of Redundant Bits per ECC CW	14	56	112	14	56	112	14	56	112	14	56	112
No. of Redundant Bits per Page	14	56	112	56	212	448	112	448	896	212	896	1792
No. of Redundant Bytes per Page	2	7	13	7	27	56	13	56	112	26	112	224

Table 88 BCH ECC HW for 1024 B Info Block Size

	Info Block Size = 1024 B											
Page Size (Bytes)	2048	2048	2048	2048	4096	4096	4096	4096	8192	8192	8192	8192
Bits Correctable per ECC CW	24	32	40	60	24	32	40	60	24	32	40	60
No. of ECC CWs per Page	2	2	2	2	4	4	4	4	8	8	8	8
log2 (Galois Field Order)	14	14	14	14	14	14	14	14	14	14	14	14
Maximum No. of Redundant Bits per ECC CW	336	448	560	840	336	448	560	840	336	448	560	840
No. of Redundant Bits per Page	627	896	1120	1680	1344	1792	2240	3360	2688	3584	4480	6720
No. of Redundant Bytes per Page	84	112	140	210	168	224	280	420	336	448	560	840

In addition, when an erased page (all contents FF) comes up for decoding, it must not report a decoding failure. This is realized with a predefined masking pattern to the ECC data.
The input parameters (register bits) are:

- Bypass (2 bits, encoder and decoder)
- ECC code size (2 bits)
- Error correction capacity (3 bits)
- Auto-detect erased pages (1 bits)
- Tolerated bit errors for erased page (3 bits)

The output parameters (register bits) are:

- Pass/fail flag (1 bit per CW)
- Number of bits corrected flag (6 bits per CW)
- Erased page detected (1 bit per CW)

6.4 Quad Serial Peripheral Interfaces

The Quad Serial Peripheral Interface (SPI) is equipped with a receive as well as a transmit FIFO with eight entries each 32-bit width. The QSPI control and status registers are explicitly accessed via the Config-Bus. The CPU handles bi-directional data traffic on the QSPI interface via another master bus access.

The QSPI supports:

- 1-bit SPI interfaces, compliance to legacy SPI standard, mode 0
- 2-bit, 4-bit mode single data rate operation
- 2-bit, 4-bit double data rate operation
- Up to 100 MHz for single/double data rate
- Adaptive delay tabs for high speed interface access timing with proven DLL designed for DDR interface
- Programming sequence to adapt different SPI flash type
- eXecution In Place, fully memory mapped access for CPU
- Auto boot support for Spansion and Macronix devices
- Embedded 256x32 bit SRAM
- Two chip selects, each can be configured to address 64 MB,128 MB range
- Baud rate generated from master `ref_clk` with programming even dividers from 2 to 32.

7 High Speed Interfaces

The high speed interfaces description covers these sections:

- **PCIe 4.0/SATA 3.2/XFI Combo Subsystem** (**Section 7.1**)
- **PCIe 3.0 Subsystem** (**Section 7.2**)
- **PON/XFI WAN Subsystem** (**Section 7.3**)
- **Ethernet PHY** (**Section 7.4**)
- **USB Subsystem** (**Section 7.5**)

7.1 PCIe 4.0/SATA 3.2/XFI Combo Subsystem

The PCIe 4.0/SATA 3.2/XFI (for HSIO1 and HSIO2) subsystem description covers these sections:

- **Overview** (**Section 7.1.1**)
- **Features** (**Section 7.1.2**)
- **Hardware Description** (**Section 7.1.3**)

The Combo-PHY subsystem includes this superset of components:

- 1 x2 SerDes PHY
- 2 SATA controller+ Physical Coding Sublayer (PCS)
- 2 PCIe Root Complex (RC) mode controller
- 2 Ethernet PCS
- 2 application configuration logic
- Multiplexing logic

7.1.1 Overview

Table 89 to **Table 90** show the overview of Combo-PHY and its application depending on the product SKU.

Note: Not all SKUs support all functions.

Table 89 Combo-PHY Overview - PCIe, SATA, and XFI Combo Mode for URX851 and URX850

Mode Number	Mode Name	PHY0	PHY1
0	Dual PCIe Single Lane	Single PCIe 4.0/3.0/2.0/1.0 RC with single lane	Single PCIe 4.0/3.0/2.0/1.0 RC with single lane
1	Single PCIe Dual Lane	Single PCIe 4.0/3.0/2.0/1.0 RC with dual lane: Aggregation	
2	RXAUI	RXAUI: Aggregation	
3	Dual XFI Single Lane	Single XFI/SGMII/USXGMII	Single XFI/SGMII/USXGMII
4	Dual SATA Single Lane	Single SATA3.2	Single SATA3.2

Table 90 Combo-PHY Overview - PCIe, SATA, and XFI Combo Mode for MxL25641

Mode Number	Mode Name	PHY0	PHY1
0	Dual PCIe Single Lane	Single PCIe 4.0/3.0/2.0/1.0 RC with single lane	Single PCIe 4.0/3.0/2.0/1.0 RC with single lane
1	Single PCIe Dual Lane	Single PCIe 4.0/3.0/2.0/1.0 RC with dual lane: Aggregation	
2	RXAUI	Not supported	
3	Dual XFI Single Lane	Single XFI/SGMII/USXGMII	Single XFI/SGMII/USXGMII
4	Dual SATA Single Lane	Not supported	

7.1.2 Features

This section provides an overview of the Combo-PHY subsystem main features.

Applications

- Single PCIe RC with dual lane
- Dual PCIe RC with a single lane each (or one lane is switched off)
- Ethernet RXAUI
- Dual Ethernet interface with a single lane each (or one lane switched off)
- Dual SATA 3.2 interface with a single lane each (or one lane switched off)

External Interfaces

- SerDes analog I/O signals per SerDes
  - Two differential data input pairs in the receive direction
  - Two differential data output pairs in the transmit direction
  - An external tuning resistor pin

Interrupt to SoC

- Active high level interrupt outputs
  - PCIe RC mode
    There are four legacy interrupts (A, B, C, D) and one overall interrupt with multiple source defined in the application logic: legacy interrupt compatible PCI mechanism.
    The EP sends in the interrupt assert message TLP which is decoded on the RC side. The RC toggles the _INTA/B/C/D pins connected to the IOAPIC.
  - The overall interrupt is the combination of the various events generated from RC itself requiring the Intel Atom processor attention, such as hardware/link ERROR, PME, VDM, TLP received/error detected.
  - This is in addition to the MSI interrupt mechanism, where EP writes MSI message over and RC directs the traffic as an MSI writes into the Intel Atom subsystem, the Intel Atom subsystem inside decodes it and directly sends to the Intel Atom processor.
  - 1 per SATA controller
  - 1 per Ethernet XPCS. The single interrupt signal connects to IOAPIC, which is various interrupt events combined from the XPCS module.

High Speed SerDes

- One instance of SerDes x2 PHY
- Complies with PCIe 1.0/1.1/2.0/3.0/4.0 specifications
  - Supports PCIe power management
  - Supports x2 to form a dual lane PCIe link
- Supports SATA 3.2 high speed serial interface
- Supports these Ethernet high speed serial interfaces:
  - 1000BASE-KX (802.3ap) and 1 Gbps SGMII [8]: 1.25 Gbaud rate
  - 10GBASE-KR: 10.3125 GT
  - 5GBASE-KR: 5.15625 GT
  - 2.5GBASE-KX and 2.5 Gbps SGMII: 3.125 GT
  - USXGMII single port: 2.578125/5.15625/10.3125 GT
- Supports programmable internal reference clock selection
  - 100 MHz reference clock for PCIe and SATA mode
  - 156.25 MHz reference clock for Ethernet mode
  - Integrated PLL for transmit clock generation
  - Receive clock recovery from the receive data
  - Clock generation for clocks required in Ethernet PCS layer

- Supports termination resistance tuning
- Supports both transmit and receive equalization and adaption
- Supports Tx amplitude control
- Supports Rx loss of signal detection
- Supports independent Rx and Tx power state controls
- Supports configuration register (CR) port (control/configuration from application logic module)
- Test and Debug Support:
  – Scan test (at speed and stuck at)
  – JTAG interface for PHY test
  – Diagnostics for characterization and ATE testing: loopback, test pattern generation, BERT, error insertion and detection

Ethernet XPCS

- Two instances of Ethernet PCS
  – Supports 8B/10B encoding (Clause 36 based)
  – Supports 64B/66B encoding (Clause 49 based)
- Supports RXAUI mode (Aggregation of two lane)
  – 10 Gbps: 6.25 GT per lane, XGMII mode
  – Supports both Dune Networks and Marvell specification
  – Applicable to only first XPCS
- Supports Clause 73 auto-negotiation in backplane Ethernet mode
  – 1000BASE-(K)X, 1 Gbps, 1.25 GT
  – 10GBASE-(K)R, 10 Gbps, 10.3125 GT
  – 5GBASE-(K)R, 5 Gbps, 5.15625 GT
  – 2.5GBASE-(K)X, 2.5 Gbps, 3.125 GT
- Supports SGMII with Clause 37
  – 1 port, 1.25GT, 8B/10B encoding
  – Supports auto-negotiation with these modes:
    1 Gbps, GMII mode
    100 Mbps, GMII mode
    10 Mbps, GMII mode
- Supports 2.5 Gbps SGMII
  – 1 port, 3.125GT, 8B/10B encoding
  – 2.5 Gbps, GMII mode
- Supports 10G-SXGMII with Clause 37
  – 1 port, 10.3125 GT, 64B/66B encoding
  – Supports auto-negotiation with these modes:
    10 Gbps, XGMII mode
    5 Gbps, XGMII mode
    2.5 Gbps, XGMII mode
    1 Gbps, GMII mode
    100 Mbps, GMII/MII mode
    10 Mbps, GMII/MII mode
- Supports 5G-SXGMII Clause 37 per Ethernet PCS
  – 1 port, 5.15625 GT, 64B/66B encoding
  – Supports auto-negotiation with these modes:
    5 Gbps, XGMII
    2.5 Gbps, XGMII
    1 Gbps, GMII
    100 Mbps, GMII/MII
    10 Mbps, GMII/MII

- Supports 2.5G-SXGMII-1p Clause 37 per Ethernet PCS
  - 1 port, 2.578125 GT, 64B/66B encoding
  - Supports auto-negotiation with these modes:
    2.5 Gbps, XGMII
    1 Gbps, GMII
    100 Mbps, GMII/MII
    10 Mbps, GMII/MII
- Clause 37 at MAC side only
- Supports Clause 72 for 10G-BASE (K)R training
- Supports forward error correction in 10G-BASE (K)R mode
- Supports EEE mode
- Supports APB for configuration

PCIe RC Mode Controller

- Two Instances of identical PCI Express root complex port
- Supports and complies with PCIe 1.0/1.1/2.0/3.0/4.0
- Supports single lane or dual lanes
- Supports 128-bit internal data-path
- Advanced power and clock management
- Internal address translation unit
- Integrated AXI bridge

Application Logic

- Two APB bridge ports: for PCIe controller, PHY configuration, XPCS configuration, and SATA controller
- Two identical instances of application logic module
- Inside the APB bridge, PCIe application logic and provides access to PHY control register indirectly
- Inside the APB bridge, access to Ethernet PCS APB configuration interface
- Supports general PCIe RC mode controller configuration and application logic
- Supports PHY configuration and status

Multiplexing Logic

- Multiplexing logic to support modes
- Multiplexing data path between PCIe RC mode controller, SATA, Ethernet PCS, and PHY
- Multiplexing control and status signals between PCIe controller, SATA, Ethernet PCS, and PHY
- Multiplexing XGMII and (G)MII interface of the XPCS

Default Mode After Reset

- The PCIe controller is always in RC mode.
- The PHY and PCIe/SATA controller/Ethernet XPCS must be enabled according to the mode configuration input before reset is released.

7.1.3 Hardware Description

This section provides a detailed functional description of the subsystem.

7.1.3.1 Top Level Block Diagram

Figure 23 shows top level block diagram and the boundaries of the subsystem in PCIe/SATA/Ethernet combo configuration.

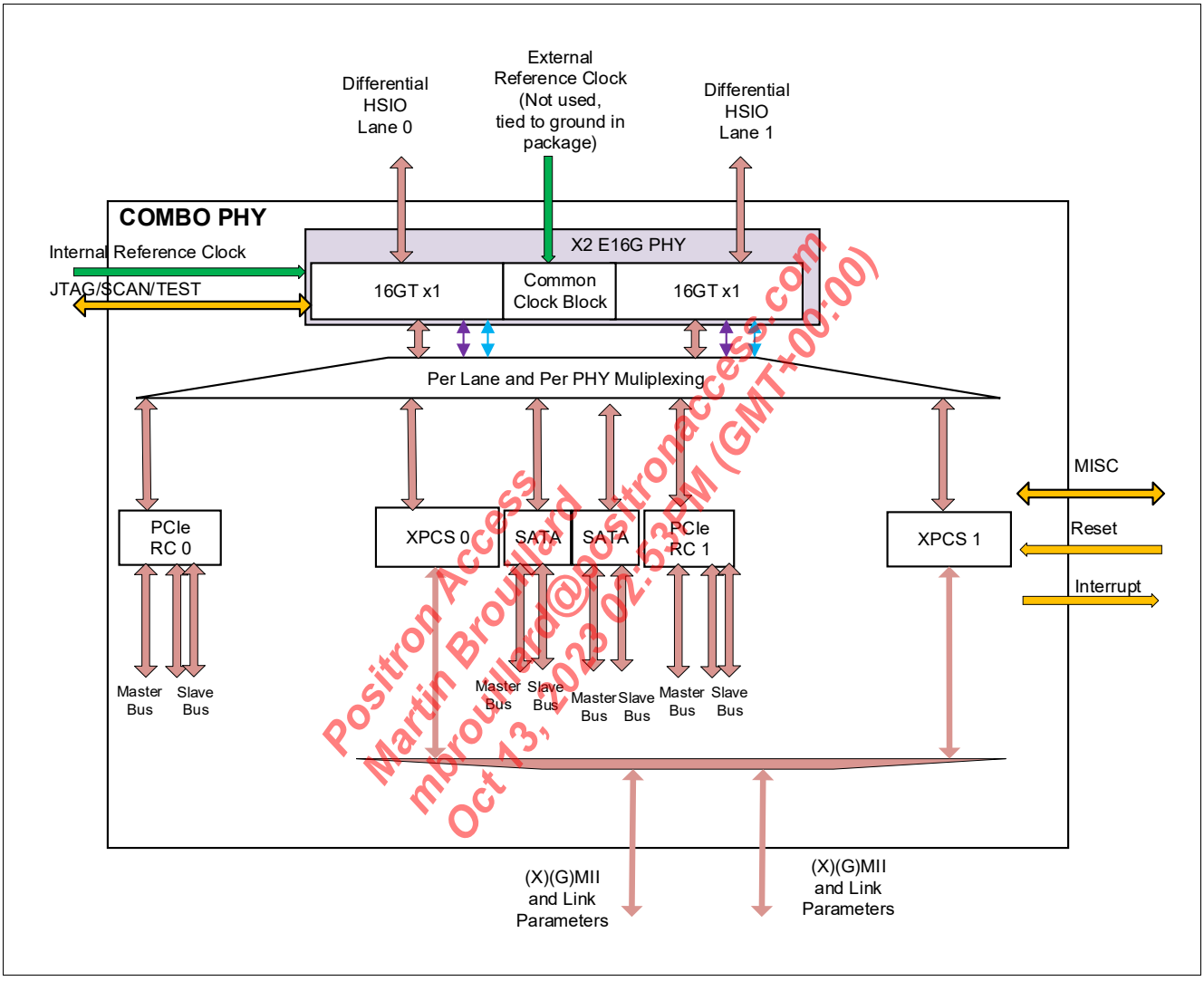


Figure 23 ComboPHY Subsystem Block Diagram

7.1.3.2 Ethernet PCS Configuration

Table 91 summarizes the first and second XPCS supported modes.

XPCS is only working at MAC side.

The mode is statically set by the configuration.

The submode and link speed are determined by auto-negotiation.

The EEE mode is supported in all modes.

Clause 72 and FEC must be supported for 10G-KR mode.

For each link, there are a separate XGMII interface and GMII/MII interface from the XPCS. A multiplexer is required to mux these two interfaces according to the selected modes.

The RXAUI mode is only applied to the first XPCS.

Table 91 Support Modes

Modes	Sub-mode Auto-negotiation	Submodes	Baud Rate	Coding	Link Speed Auto-negotiation	Link Speed	XGMII /GMII/MII Modes
RXAUI	No	RXAUI	6.25 GT per lane	8b/10b	No	10 Gbps	XGMII
Backplane Ethernet	Clause 73	10G-(K)R	10.3125 GT	64b/66b	No	10 Gbps	XGMII
		5G-(K)R	5.15625 GT	64b/66b	No	5 Gbps	XGMII
		2.5G-(K)X	3.125 GT	8b/10b	No	2.5 Gbps	XGMII
		1000-(K)X	1.25 GT	8b/10b	No	1 Gbps	GMII
SGMII-1G	No	SGMII-1G	1.25 GT	8b/10b	Clause 37	1 Gbps, 100 Mbps, 10 Mbps	GMII, GMII/MII, GMII/MII
SGMII-2.5G	No	SGMII-2.5G	3.125 GT	8b/10b	No	2.5 Gbps	GMII
10G-SXGMII (1 Port)	No	10G-SXGMII	10.3125 GT	64b/66b	Clause 37	10 Gbps, 5 Gbps, 2.5 Gbps, 1 Gbps, 100 Mbps, 10 Mbps	XGMII, XGMII, XGMII, GMII, GMII/MII, GMII/MII
5G-SXGMII (1 Port)	No	5G-SXGMII	5.15625 GT	64b/66b	Clause 37	5 Gbps, 2.5 Gbps, 1 Gbps, 100 Mbps, 10 Mbps	XGMII, XGMII, GMII, GMII/MII, GMII/MII
2.5G-SXGMII (1 Port)	No	2.5G-SXGMII	2.578125 GT	64b/66b	Clause 37	2.5 Gbps, 1 Gbps, 100 Mbps, 10 Mbps	XGMII, GMII, GMII/MII, GMII/MII

7.1.3.3 Multiplexing Logic Applicable for URX851/URX850

7.1.3.3.1 Single PCIe RC Dual Lane Mode

Figure 24 shows the diagram for single PCIe dual lane mode in PCIe/Ethernet combo configuration.

- The PHY is operating in x2 mode.
- PCIe controller 0 dual lanes are connected to each lane of the x2 PHY. RC mode only. It supports PCIe 1.0/1.1/2.0/3.0/4.0.
- Application logic module 0 (both PHY configuration/status and CR port) is connected to RC 0 and PHY 0.
- PCIe controller 1 is not used and is disabled.
- Application logic module 1 is connected to RC1.

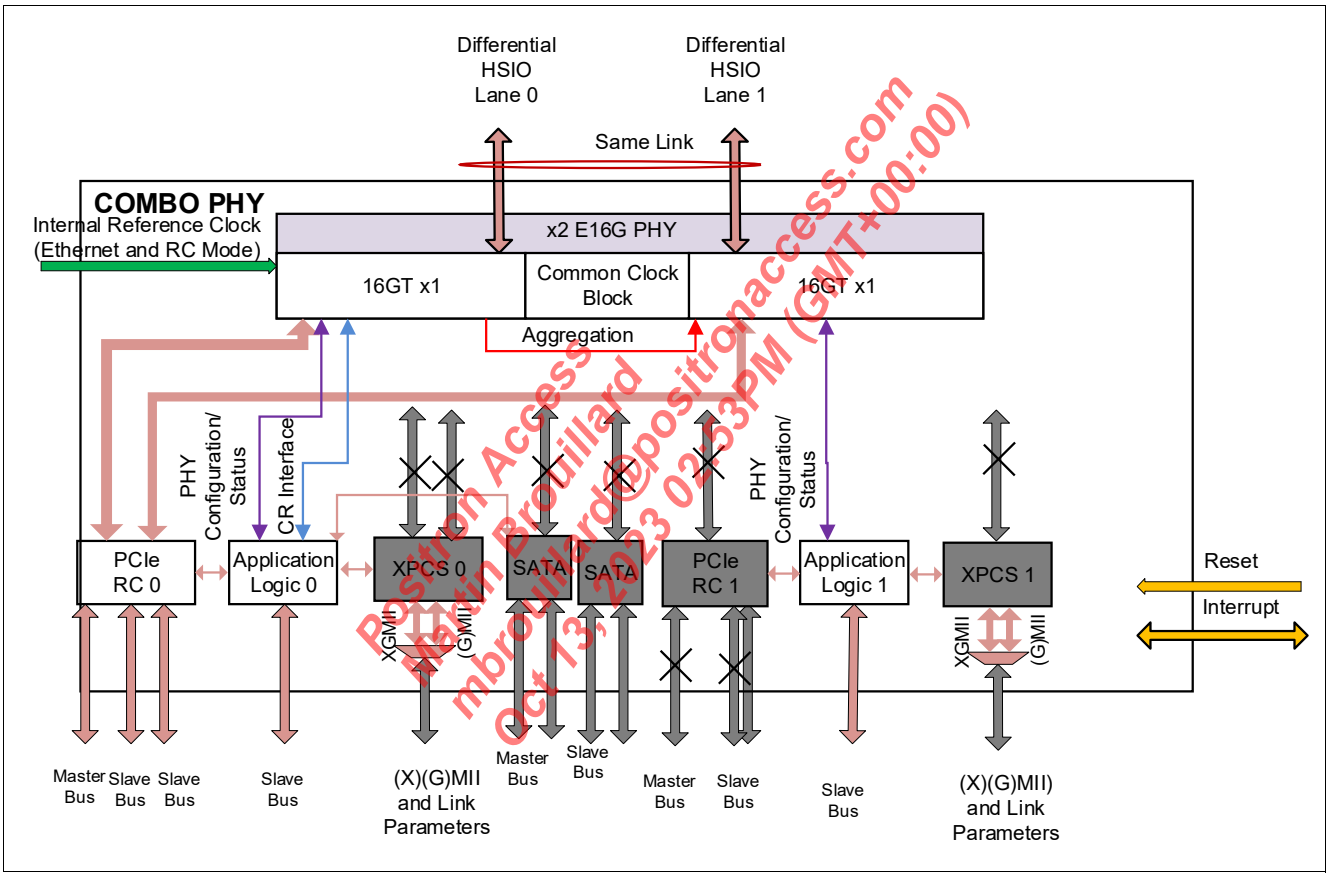


Figure 24 Single PCIe Dual Lane (in PCIe/SATA/Ethernet Combo Configuration)

7.1.3.3.2 Dual PCIe RC Single Lane Mode

Figure 25 shows the diagram for dual PCIe RC single lane mode in PCIe only configuration.

- The two lanes of the PHY are in bifurcation mode.
- PCIe controller 0 single lane is connected to lane 0 of the PHY. It supports PCIe 1.0/1.1/2.0/3.0/4.0.
- Application logic module 0 (both PHY configuration/status and CR port) is connected to the PHY.
- PCIe controller 1 single lane is connected to lane 1 of the PHY. It supports PCIe 1.0/1.1/2.0/3.0/4.0.
- Application logic module 1 is connected to PHY0 Lane 1.

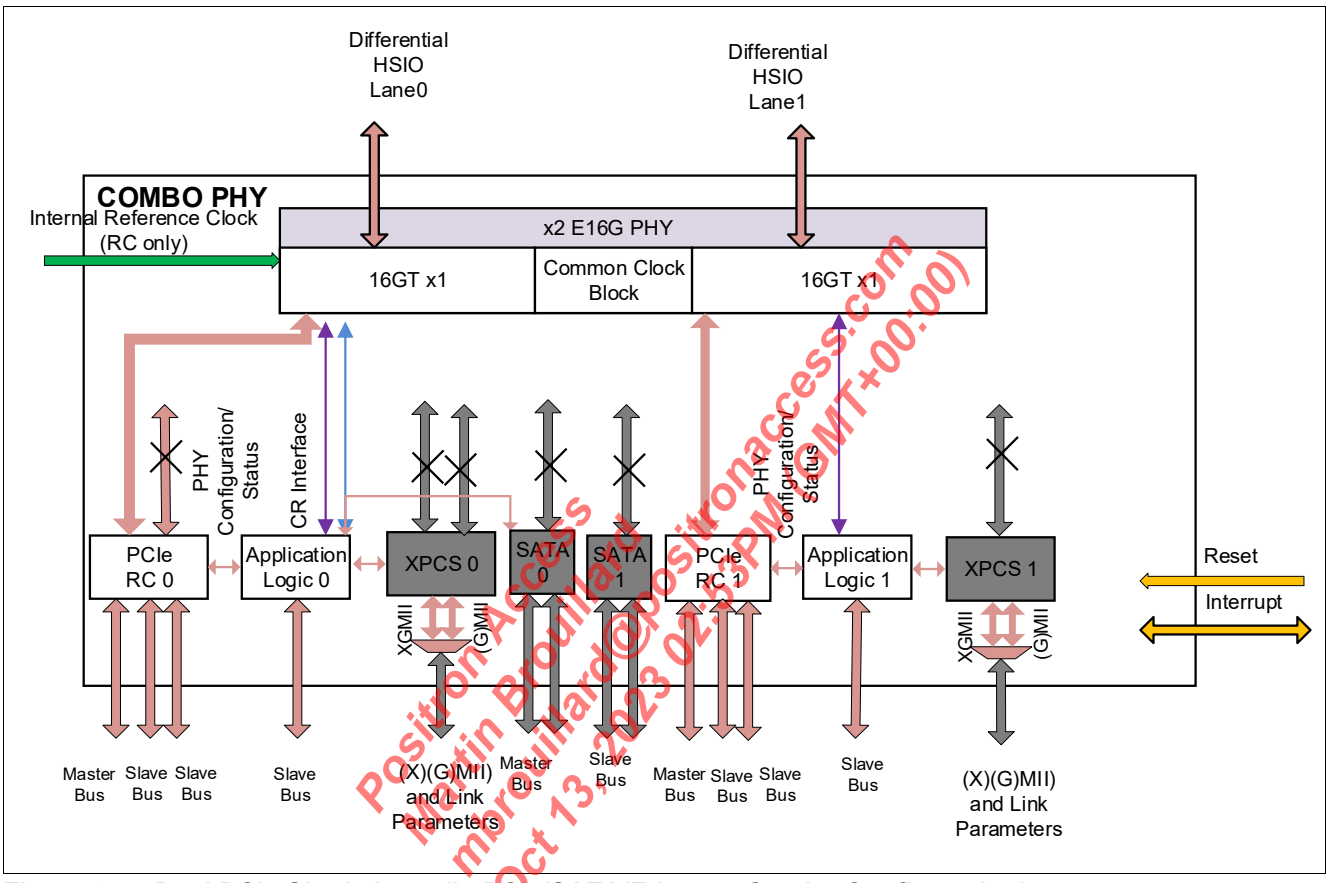


Figure 25 Dual PCIe Single Lane (in PCIe/SATA/Ethernet Combo Configuration)

7.1.3.3.3 RXAUI Dual Lane Mode

This mode is valid for PCIe/Ethernet combo configuration.

Figure 26 shows the diagram for RXAUI dual lane mode in PCIe/Ethernet Combo configuration.

- The PHY is operating in x2 mode.
- The dual lane of XPCS0 are connected to each lane of the two PHY. XPCS is working in RXAUI mode.
- XPCS1's lane is not connected.
- Application logic module 0 (PHY configuration/status and CR port) is connected to PHY0.
- PCIe controllers 0 and 1 are not used and are disabled.
- Application logic module 1 is connected to PHY0 Lane 1.

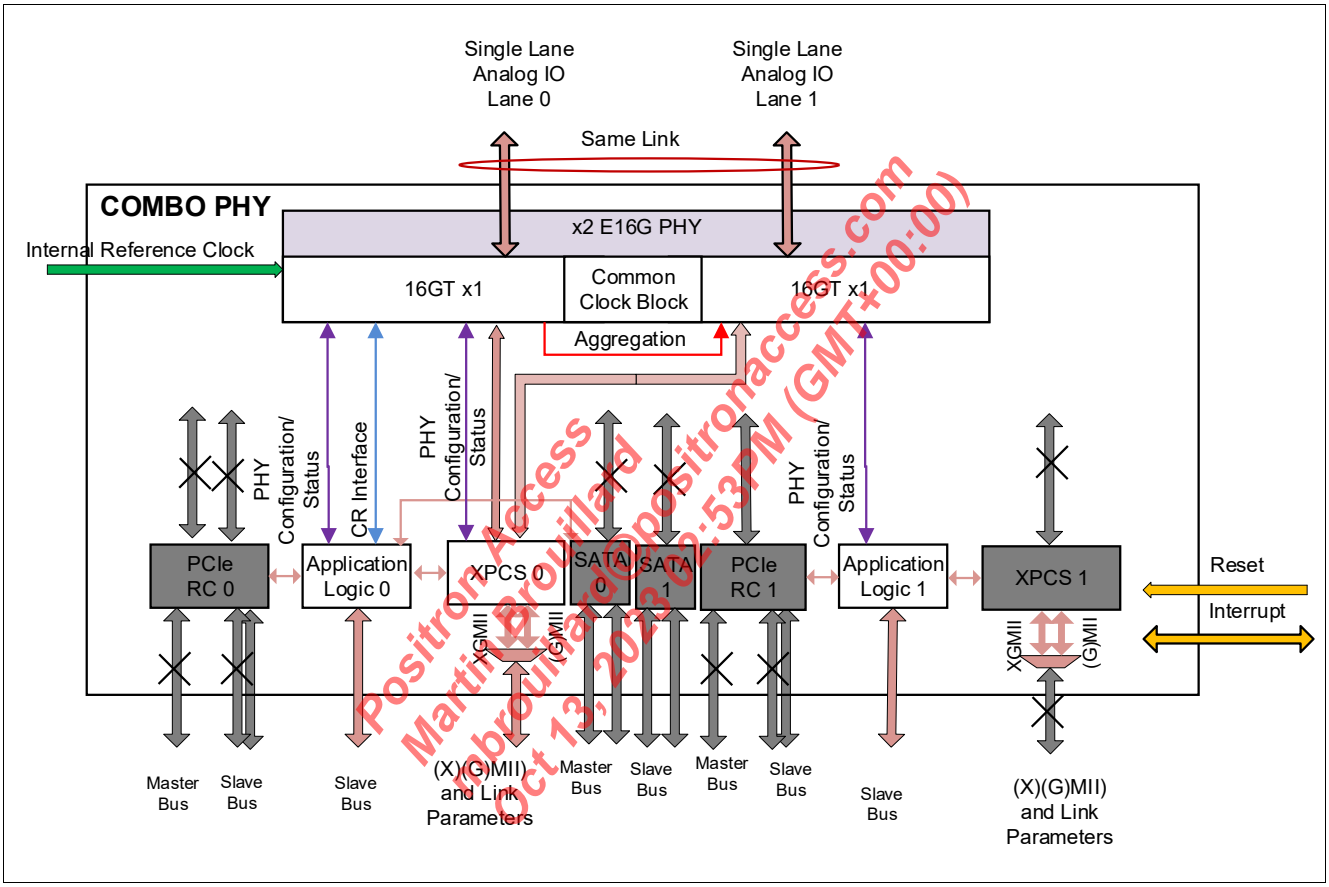


Figure 26 RXAUI Dual Lane

7.1.3.3.4 Dual Ethernet Single Lane Mode

This mode is valid for PCIe/Ethernet combo configuration.

Figure 27 shows the diagram for dual Ethernet single lane mode in PCIe/Ethernet Combo configuration. The two Ethernet links work independently.

- PCIe controllers 0 and 1 are not connected.
- Application logic module 0 (both PHY configuration/status and CR port) is connected to PHY Lane 0.
- Application logic module 1 is connected to PHY Lane 1.
- XPCS0's lane 0 is connected to PHY Lane 0. XPCS must operate in these modes:
  - 1000BASE-X/SGMII
  - 2.5G SGMII
  - USXGMII-1port (10G-SXGMII, 5G-SXGMII, and 2.5G-SXGMII)
  - Backplane Ethernet mode (10GBASE-(K)R, 5GBASE-(K)R, 2.5GBASE-(K)X, and 1GBASE-(K)X)
- The lane 0 of XPCS1 is connected to PHY LANE 1. XPCS must operate in these modes:
  - 1000BASE-X/SGMII
  - 2.5G SGMII
  - USXGMII-1-port (10G-SXGMII, 5G-SXGMII, and 2.5G-SXGMII)
  - Backplane Ethernet mode (10GBASE-(K)R, 5GBASE-(K)R, 2.5GBASE-(K)X, and 1GBASE-(K)X)

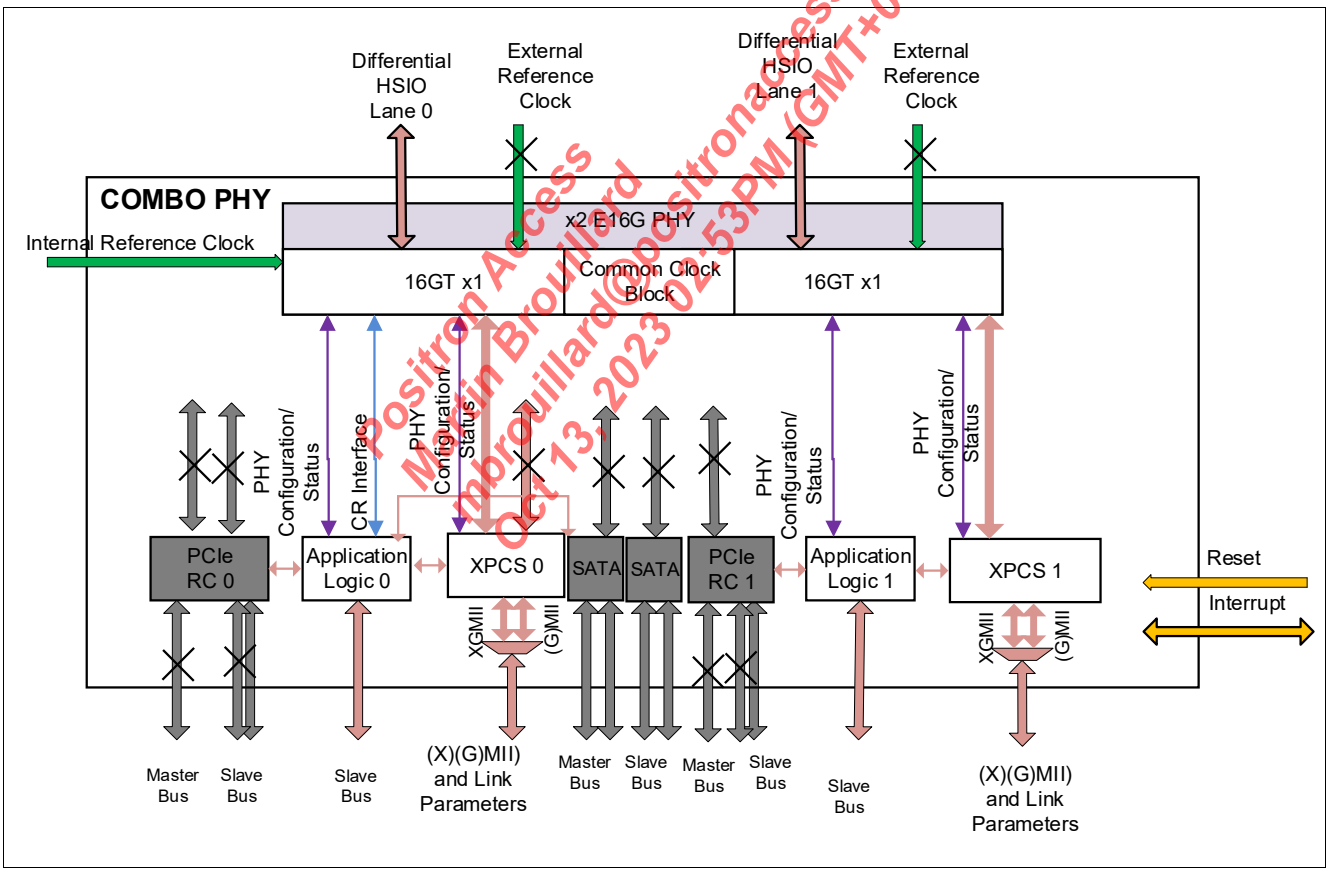


Figure 27 Dual Ethernet Single Lane

7.1.3.3.5 SATA Mode

This mode is valid for PCIe/SATA/Ethernet combo configuration.

Figure 28 shows the diagram for dual Ethernet single lane mode in PCIe/SATA/Ethernet Combo configuration.

- PCIe controllers 0 and 1 are not connected.
- XPCS0 and 1 are not connected.
- Application logic module 0 (both PHY configuration/status and CR port) is connected to PHY Lane 0.
- Application logic module 1 is connected to PHY Lane 1.
- Lane 0 of SATA controller is connected to lane 0 of PHY0. It supports SATA 3.0/3.1/3.2.
- Lane 1 of SATA controller is connected to lane 1 of PHY0. It supports SATA 3.0/3.1/3.2.

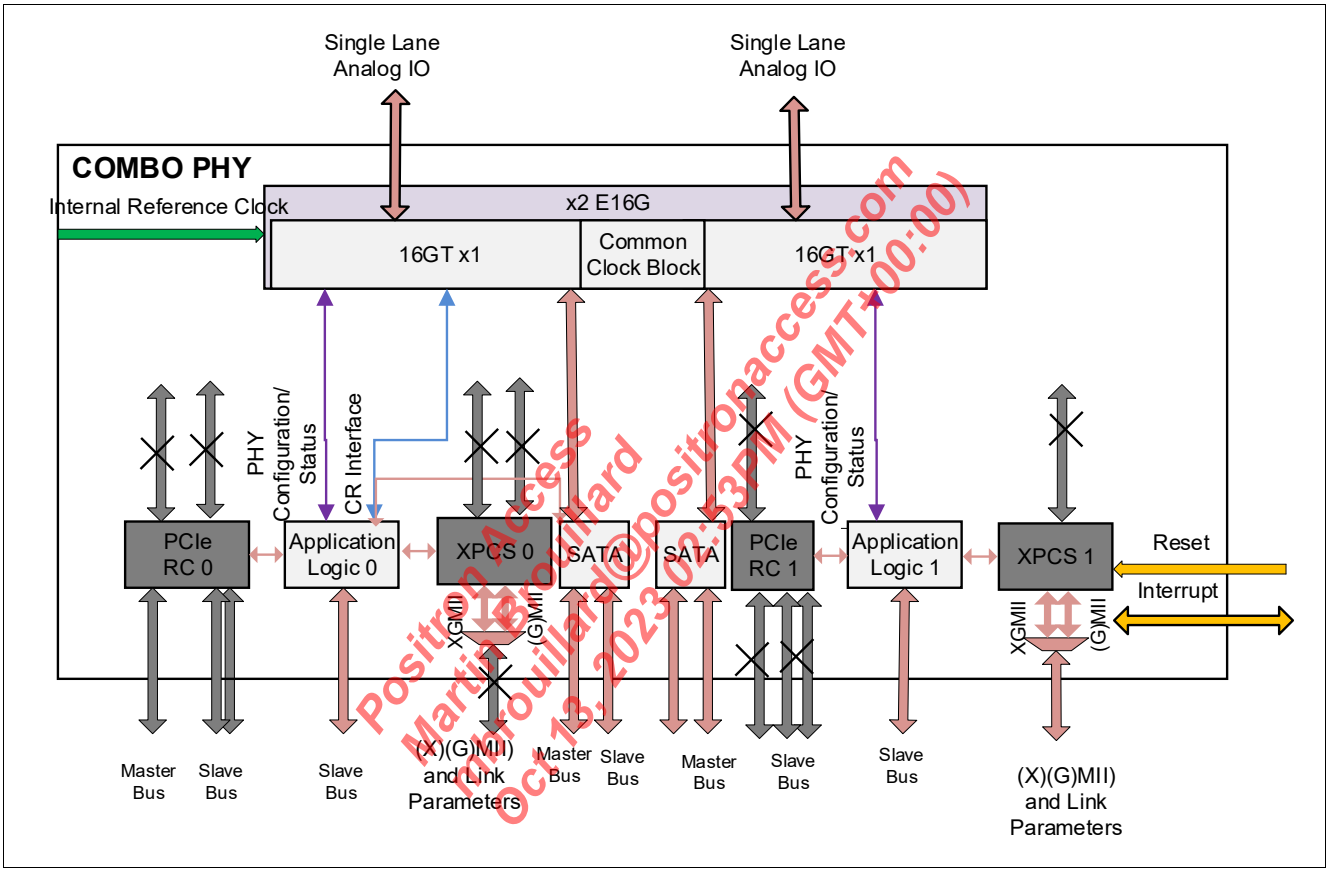


Figure 28 SATA Mode

7.1.3.4 Multiplexing Logic Applicable for MxL25641

7.1.3.4.1 Single PCIe RC Dual Lane Mode

Figure 24 shows the diagram for single PCIe dual lane mode in PCIe/Ethernet combo configuration.

- The PHY operates in x2 mode.
- PCIe controller 0 dual lanes is connected to each lane of the x2 PHY. RC mode only. It supports PCIe 1.0/1.1/2.0/3.0/4.0.
- Application logic module 0 (both PHY configuration/status and CR port) is connected to RC 0 and PHY 0.
- PCIe controller 1 is not used and is disabled.
- Application logic module 1 is connected to RC1.

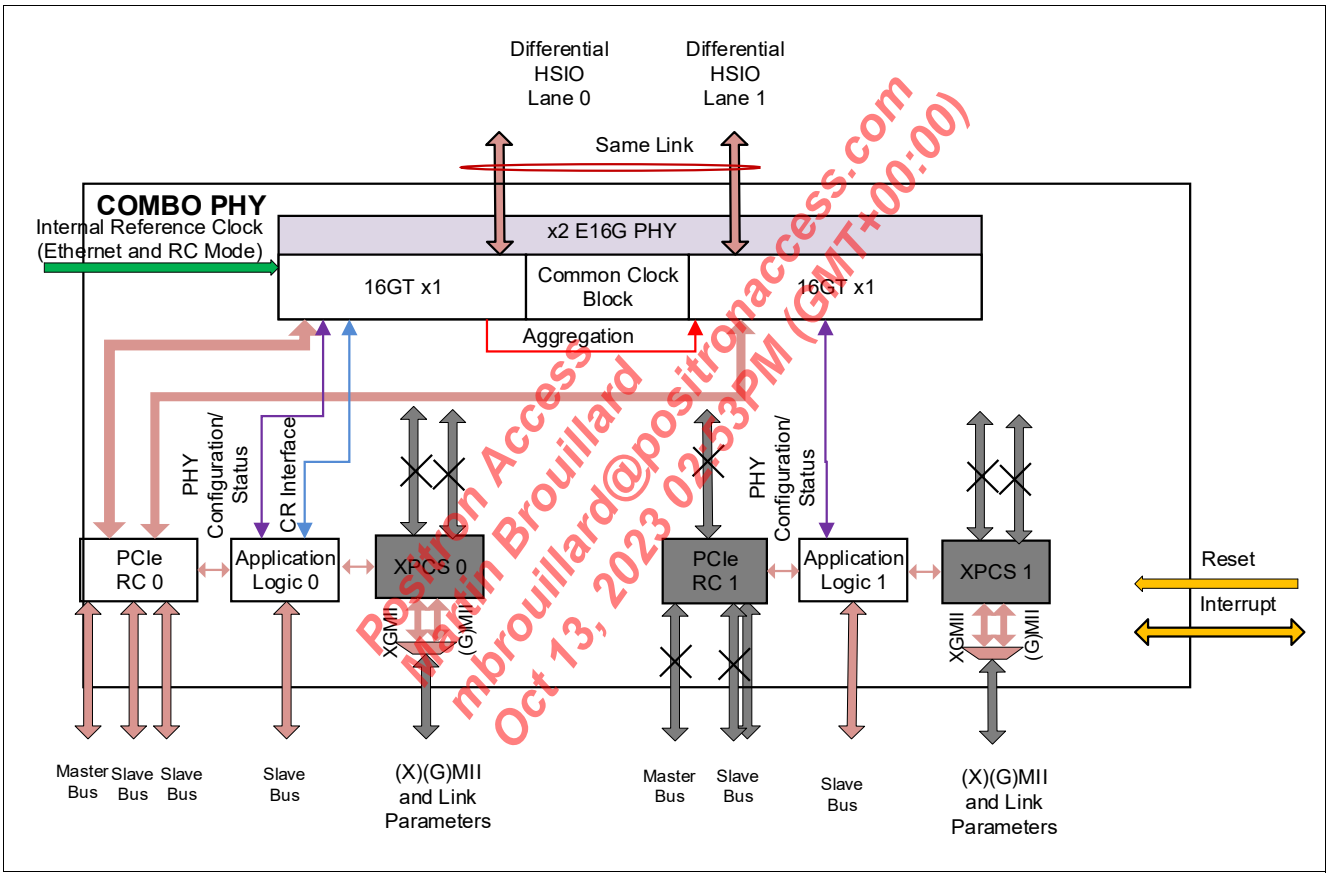


Figure 29 Single PCIe Dual Lane (in PCIe/Ethernet Combo Configuration)

7.1.3.4.2 Dual PCIe RC Single Lane Mode

Figure 25 shows the diagram for dual PCIe RC single lane mode in PCIe only configuration.

- The two lanes of the PHY are in bifurcation mode.
- PCIe controller 0 single lane is connected to lane 0 of the PHY. It supports PCIe 1.0/1.1/2.0/3.0/4.0.
- Application logic module 0 (both PHY configuration/status and CR port) is connected to PHY.
- PCIe controller 1 single lane is connected to lane 1 of the PHY. It supports PCIe 1.0/1.1/2.0/3.0/4.0.
- Application logic module 1 is connected to PHY0 Lane 1.

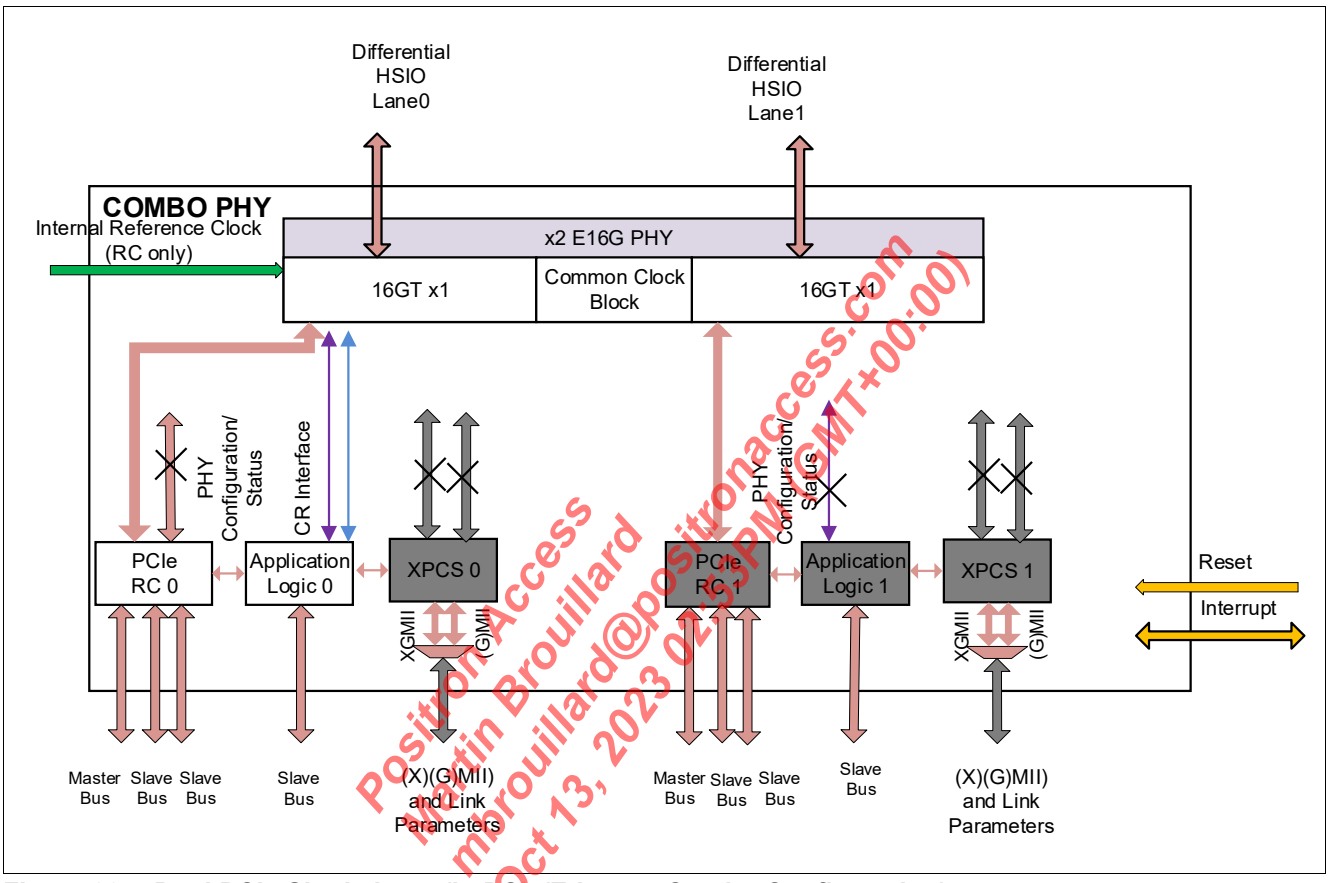


Figure 30 Dual PCIe Single Lane (in PCIe/Ethernet Combo Configuration)

7.1.3.4.3 Dual Ethernet Single Lane Mode

This mode is valid for PCIe/Ethernet combo configuration.

Figure 31 shows the diagram for dual Ethernet single lane mode in PCIe/Ethernet Combo configuration. The two Ethernet links work independently.

- PCIe controllers 0 and 1 are not connected.
- Application logic module 0 (both PHY configuration/status and CR port) is connected to PHY Lane 0.
- Application logic module 1 is connected to PHY Lane 1.
- XPCS0's lane 0 is connected to PHY Lane 0. XPCS must operate in these modes:
  - 1000BASE-X/SGMII
  - 2.5G SGMII
  - USXGMII-1port (10G-SXGMII, 5G-SXGMII, and 2.5G-SXGMII)
  - Backplane Ethernet mode (10GBASE-(K)R, 5GBASE-(K)R, 2.5GBASE-(K)X, and 1GBASE-(K)X)
- The lane 0 of XPCS1 is connected to PHY LANE 1. XPCS must operate in these modes:
  - 1000BASE-X/SGMII
  - 2.5G SGMII
  - USXGMII-1-port (10G-SXGMII, 5G-SXGMII, 2.5G-SXGMII)
  - Backplane Ethernet mode (10GBASE-(K)R, 5GBASE-(K)R, 2.5GBASE-(K)X, and 1GBASE-(K)X)

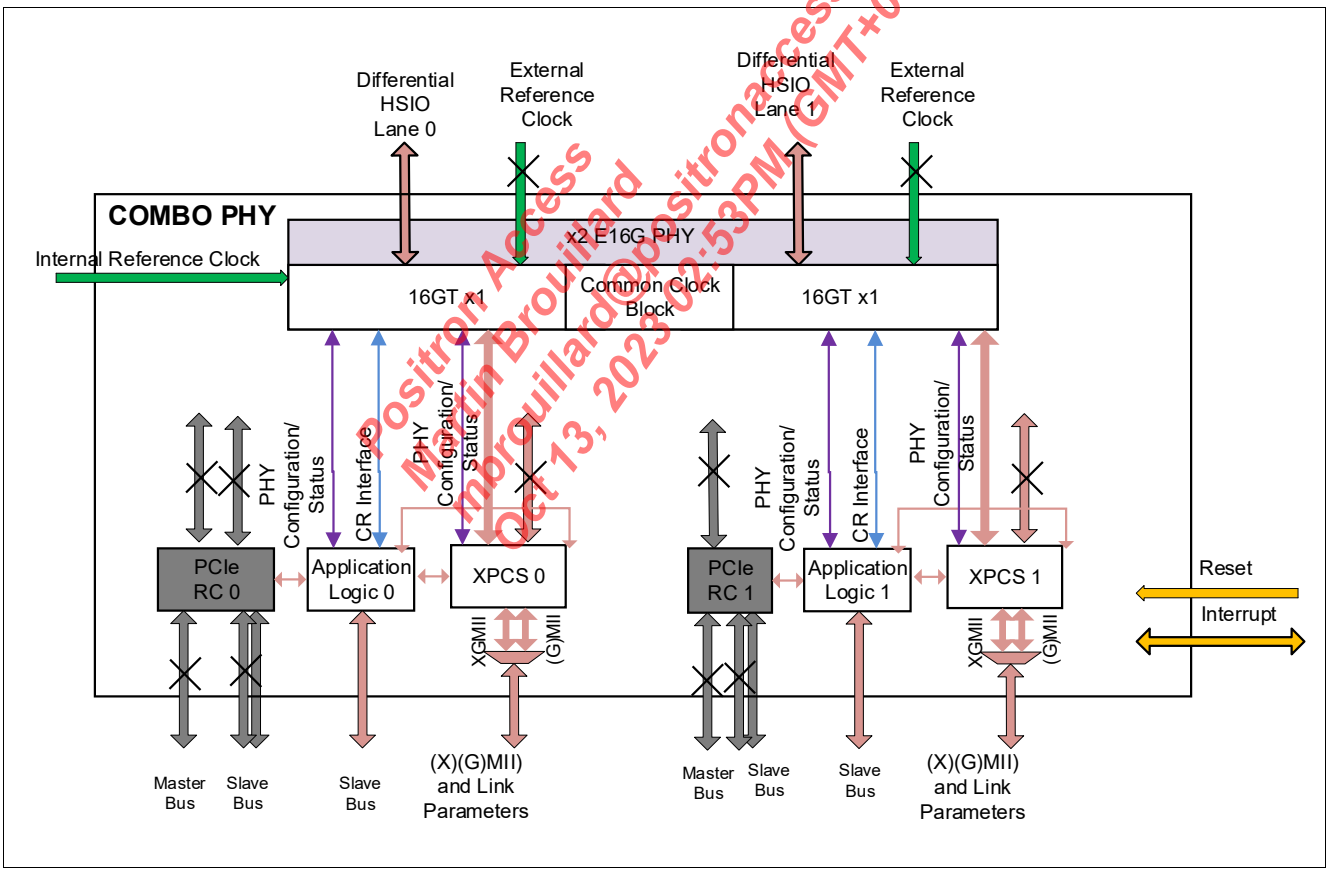


Figure 31 Dual Ethernet Single Lane

7.1.3.5 PCIe RC Mode Reference Clock Output

For each PCIe SerDes, there is a pair of reference common-mode logic (CML) clock pad (outside of the ComboPHY subsystem).

Figure 32 shows the integration of the PCIe reference clock output.

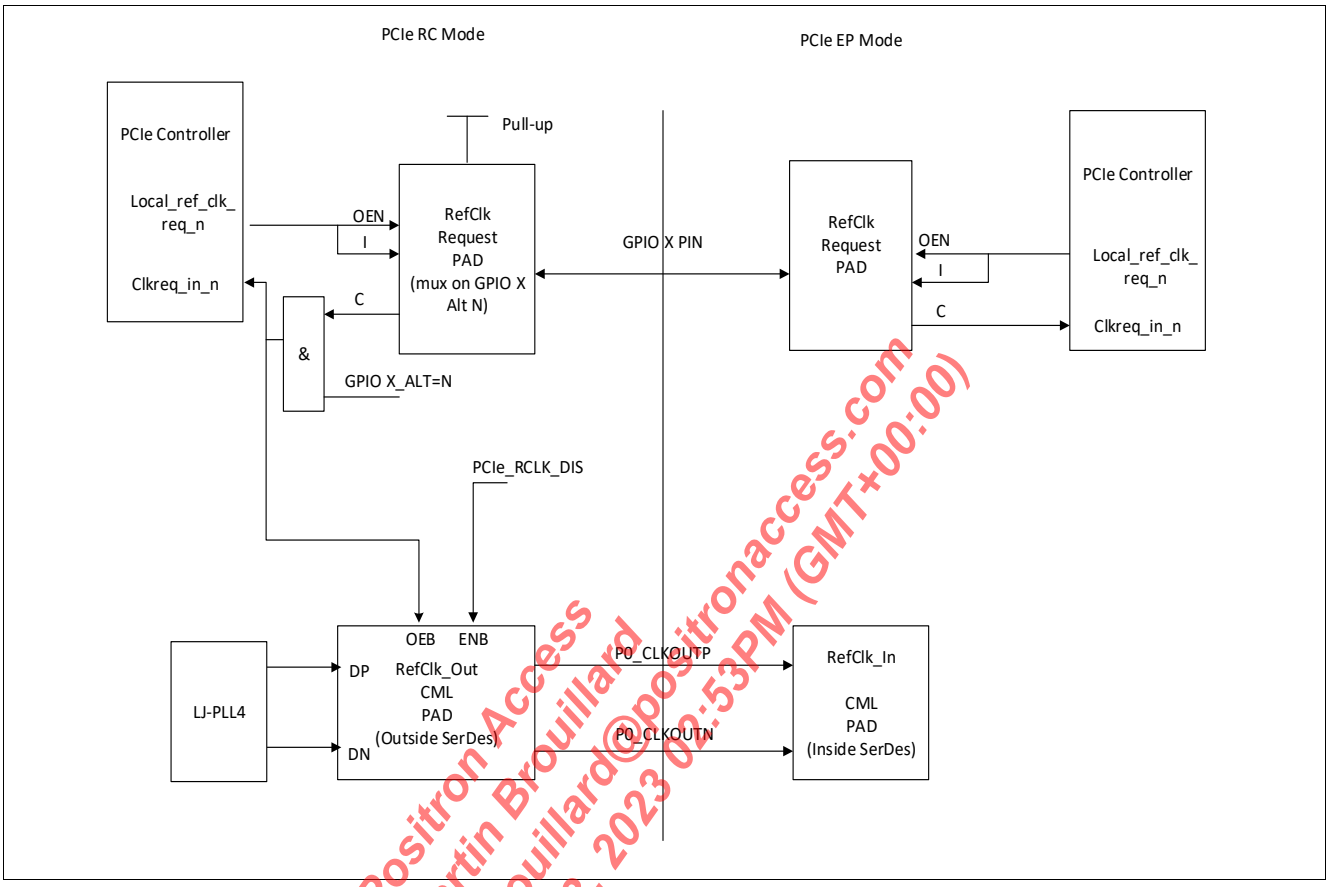


Figure 32 PCIe Reference Clock Connection

7.1.3.6 Clock Gating

This section describes the clock gating functionality.

Subsystem Static Disable Mode

When the whole subsystem is disabled (PCIe 0,1 controller clock disable, XPCS0, 1 clock disable, SATA 0, 1 clock disable via chip top register), all the clocks (differential reference clock, NoC interface clock, EEE clock) are switched off and the whole subsystem is configured to the reset state.

Sub-module Mode

Each sub-module (XPCS0, XPCS1, PCIe0, PCIe1, SATA 0, SATA 1, PHY) supports the disable mode. When the module is disabled, all the input clocks to that module are switched off and the module is configured to the reset state. There is no direct PHY clock disable control. This is controlled by XPCS0/1, PCIe 0/1 clock disable, SATA clock disable and ComboPHY mode indirectly.

1. Combo internal clocks (for example `pclk`) - gated by CGC signals from CGU (software) and comboPHY mode (hardware).
2. Combo outside clocks (for example `axi` clocks) - gated by CGC signals only (software).

Low Power Idle Clock Stop

XGMII/GMII/MII Rx interface clock (to GSWIP-O MAC and to XPCS) is switched off in EEE LPI mode and the control is from XPCS.

XGMII/GMII/MII Tx interface clock (to GSWIP-O MAC and to XPCS) is switched off in EEE LPI mode and the control is from GSWIP-O.

7.2 PCIe 3.0 Subsystem

The PCIe 3.0 subsystem description covers these sections:

- **Overview** (**Section 7.2.1**)
- **Features** (**Section 7.2.2**)
- **Hardware Description** (**Section 7.2.3**)

The PCIe subsystem includes these super-set of components:

- 1 SerDes x2 PHY
- 2 PCIe RC mode controller
- 2 application configuration logic
- Multiplexing logic

7.2.1 Overview

Table 92 gives an overview of PCIe PHY and its application.

Attention: MxL25641 does not support the PCIe 3.0 subsystem.

Table 92 PCIe PHY Overview

Mode Number	Mode Name	Product	SerDes	PHY0	PHY1
0	Dual PCIe Single Lane	URX851, URX850	SerDes x2	Single PCIe 3.0/2.0/1.0 RC with single lane	Single PCIe 3.0/2.0/1.0 RC with single lane
1	Single PCIe Dual Lane	URX851, URX850		Single PCIe 3.0/2.0/1.0 RC with dual lane: Aggregation. Must be controller 0.	

7.2.2 Features

This section provides an overview of the PCIe PHY subsystem main features.

Applications

- Single PCIe RC with dual lane
- Dual PCIe RC with single lane each
- Single PCIe RC with single lane

External Interfaces

- SerDes analog I/O signals
  - Single PHY providing x2 lanes (internal, this is two SerDes running at aggregation mode)

Memory Interfaces

- Memories per PCIe controller

Interrupt to/from SoC

- Active high level interrupt outputs
  - PCIe RC mode: Four legacy interrupt (A, B, C, D) and one overall interrupt with multiple source defined in application logic.

High Speed SerDes

- One instance of SerDes x2 PHY
- Complies with PCIe 1.0/1.1/2.0/2.1/3.0 specifications

  - Support PCIe power management
  - Support aggregation of two separate x1 PHY to form a dual lane PCIe link
- Support programmable internal or external reference clock selection
  - 100 MHz reference clock for PCIe mode
  - Integrated PLL for transmit clock generation
  - Receive clock recovery from the receive data
- Support termination resistance tuning
- Support both transmit and receive equalization and adaption
- Support Tx amplitude control
- Support Rx loss of signal detection
- Support independent Rx and Tx power state controls
- Support configuration register (CR) port (control/configuration from application logic module)

PCIe RC Mode Controller

- Two instances of PCIe controllers
  - Identification configuration
  - in system application, controller 0 supports dual/single lane mode; controller 1 supports single lane mode
- Supports only PCIe RC mode
- Supports and complies with PCIe 1.0/1.1/2.0/2.1/3.0
- Supports 128 bit internal data-path
- Advanced power and clock management
- Internal address translation unit

Application Logic

- Two APB bridge ports: for PCIe controller and PHY configuration
- Two instances of application logic module
- Supports general PCIe dual mode controller configuration and application logic
- Supports PHY configuration and status

Multiplexing Logic

- Multiplexing logic to support modes defined in Table 92
- Multiplexing data path between PCIe dual mode controller and PHY
- Multiplexing control and status signals between PCIe dual mode controller and PHY

Default Mode After Reset

- The default mode for PCIe 0/1 controller is RC mode
- PHY and PCIe controller must be enabled according to the mode configuration input prior to reset for the PHY, after reset for controller.

7.2.3 Hardware Description

This section provides a detailed functional description of the subsystem.

7.2.3.1 Top Level Block Diagram

Figure 33 shows the top level block diagram and the boundaries of the subsystem in PCIe configuration.

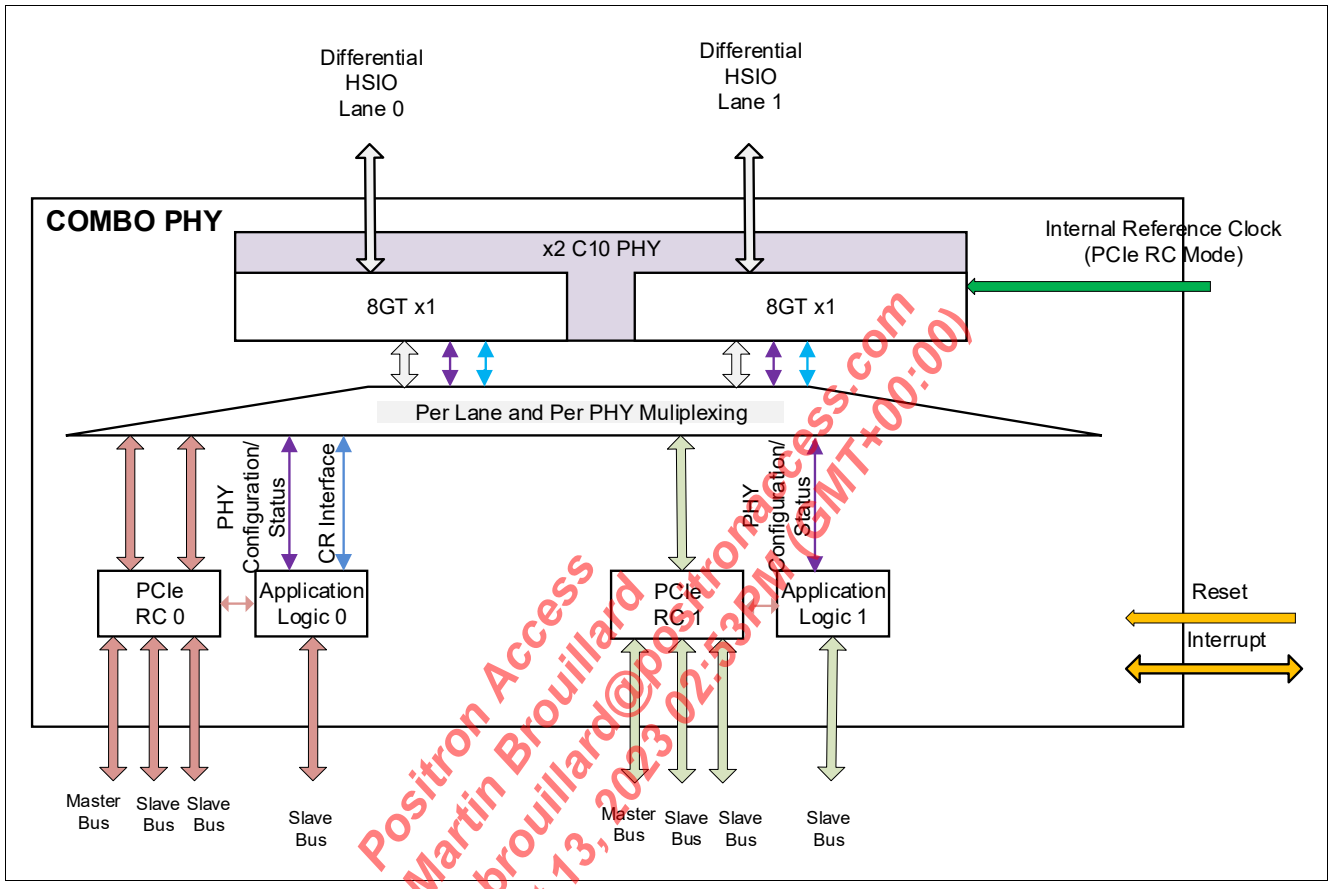


Figure 33 PCIe 3.0 Subsystem Block Diagram (PCIe Configuration Mode)

7.2.3.2 PCIe Dual Mode Controller Configuration

See Section 7.2.3.4 and Section 7.2.3.5 for a description of the PCIe dual mode configuration.

7.2.3.3 Application Logic Module

Figure 34 shows the block diagram of application logic module.

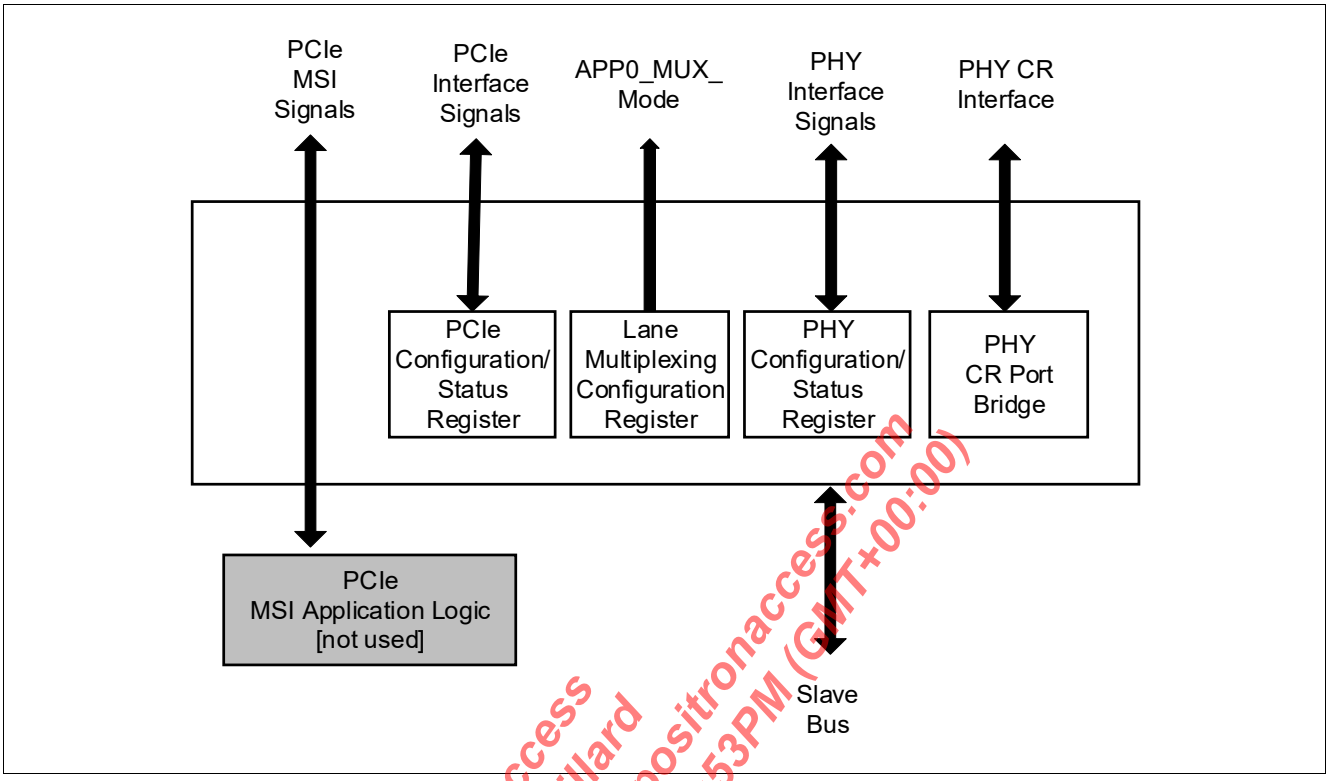


Figure 34 Application Logic Block Diagram

PHY CR Port Bridge

This module is responsible for bridging from a common APB slave interface to PHY CR (control register) port.

PHY Configuration/Status Registers

This module implements PHY configuration/status registers. CPU can access these registers via APB slave interface. The outputs of the registers are connected to PHY interface configuration signals. The PHY interface status signals are connected to the inputs of the registers.

PCIe Configuration/Status Registers

This module implements PCIe configuration/status registers. CPU can access these registers via APB slave interface. The outputs of the registers are connected to PCIe interface configuration signals. The PCIe interface status signals are connected to the inputs of the registers.

7.2.3.4 Single PCIe RC Dual Lane Mode

Figure 35 shows the diagram for the single PCIe dual lane mode in PCIe combo configuration.

- The PHY is operating in x2 mode.
- PCIe controller 0 dual lanes is connected to each lane of the x2 PHY. It supports PCIe 1.0/1.1/2.0/2.1/3.0.
- Application logic module 0 (PHY configuration/status and CR port) is connected to PHY Lane 0.
- PCIe controller 1 is not used and is disabled.
- Application logic module 1 (PHY configuration per status) is connected to PHY Lane 1, not in use.

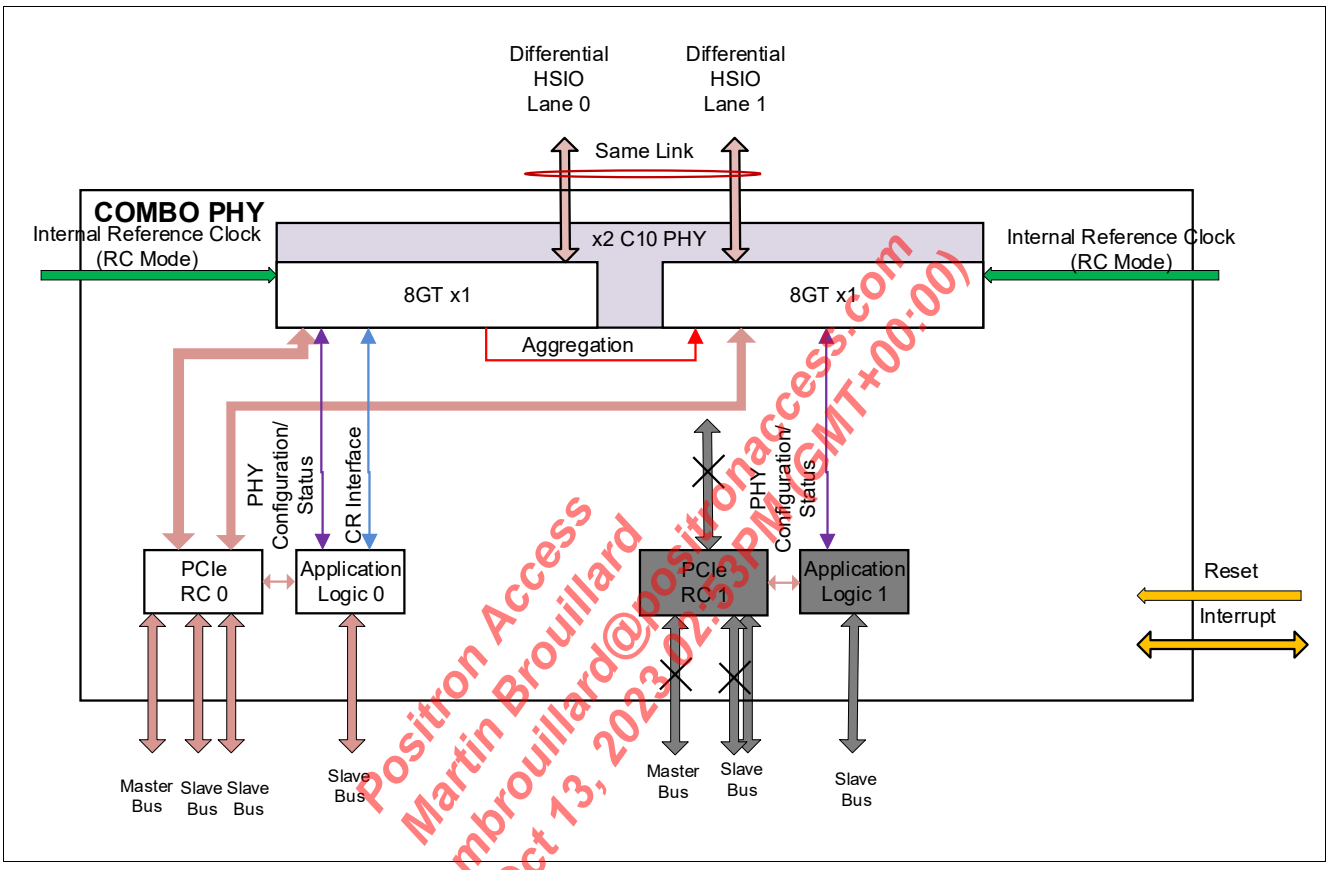


Figure 35 Single PCIe Dual Lane

7.2.3.5 Dual PCIe RC Single Lane Mode

Figure 36 shows the diagram for the dual PCIe RC single lane mode in PCIe only configuration.

- The two PHYs are in bifurcation mode.
- PCIe controller 0 single lane is connected to lane 0 of the PHY, the second lane link is not connected. It supports PCIe 1.0/1.1/2.0/2.1/3.0.
- Application logic module 0 (both PHY configuration/status and CR port) is connected to the PHY.
- PCIe controller 1 single lane is connected to lane 1 of the PHY. It supports PCIe 1.0/1.1/2.0/2.1/3.0.
- Application logic module 1 is connected to PHY Lane 1.

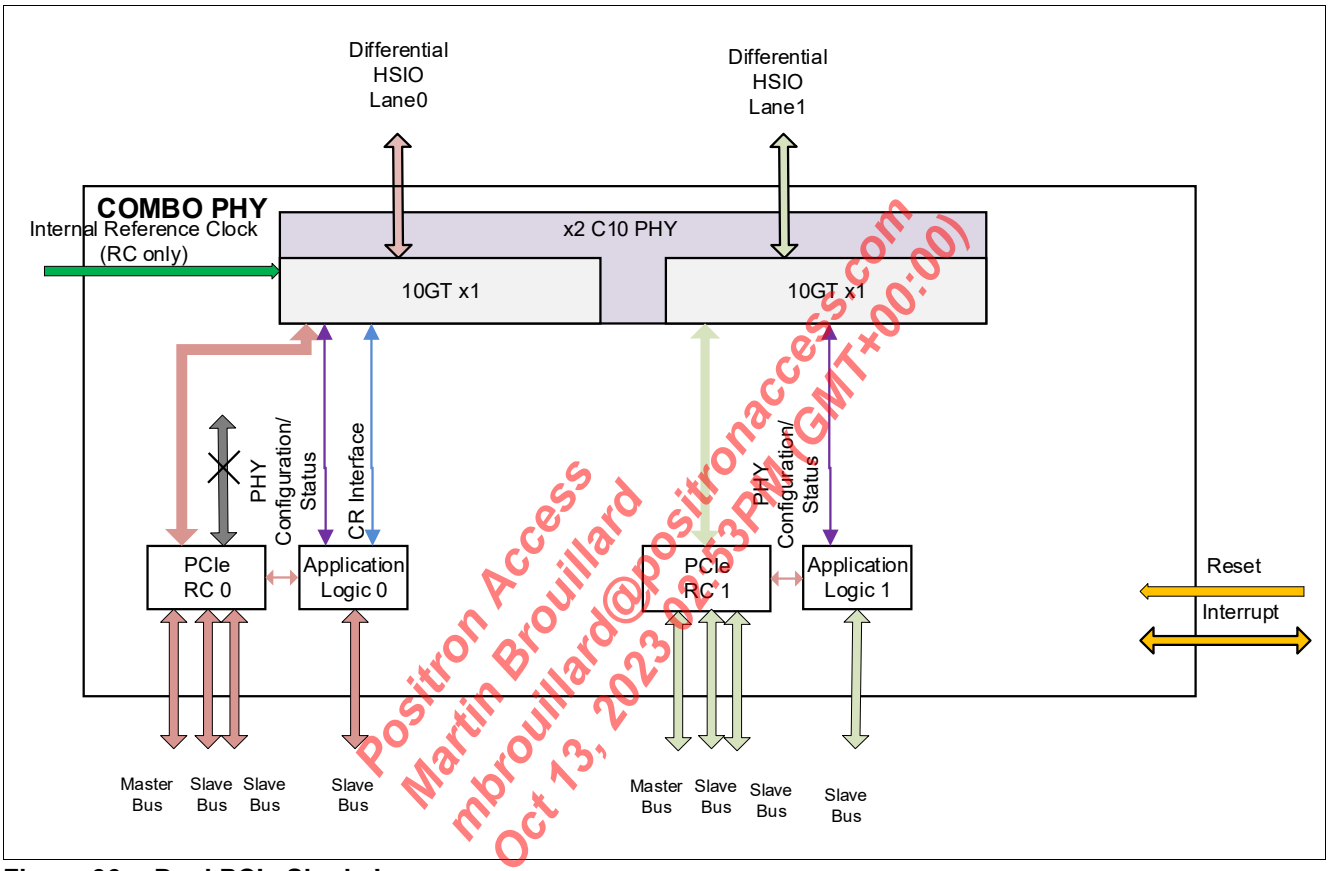


Figure 36 Dual PCIe Single Lane

7.2.3.6 PCIe RC Mode Reference Clock Output

For each PCIe SerDes, there is a pair of reference CML clock pad (outside of the Combo PHY subsystem).

Figure 32 shows the integration of PCIe reference clock output.

7.3 PON/XFI WAN Subsystem

PON/XFI WAN subsystem includes these components:

- PON/XFI WAN interface module
- GSWIP-O module

7.3.1 PON/XFI WAN Interface Module

PON/XFI WAN interface module includes these components:

- 1 instance of 16G x1 PHY
- 1 instance of Ethernet PCS
- 1 instance of PON-IP, including memories. For URX851 and MxL25641 only.
- Application logic
- Multiplexing and glue logic

7.3.1.1 Overview

Figure 37 shows the block diagram of PON/XFI WAN interface module in URX851 and MxL25641.

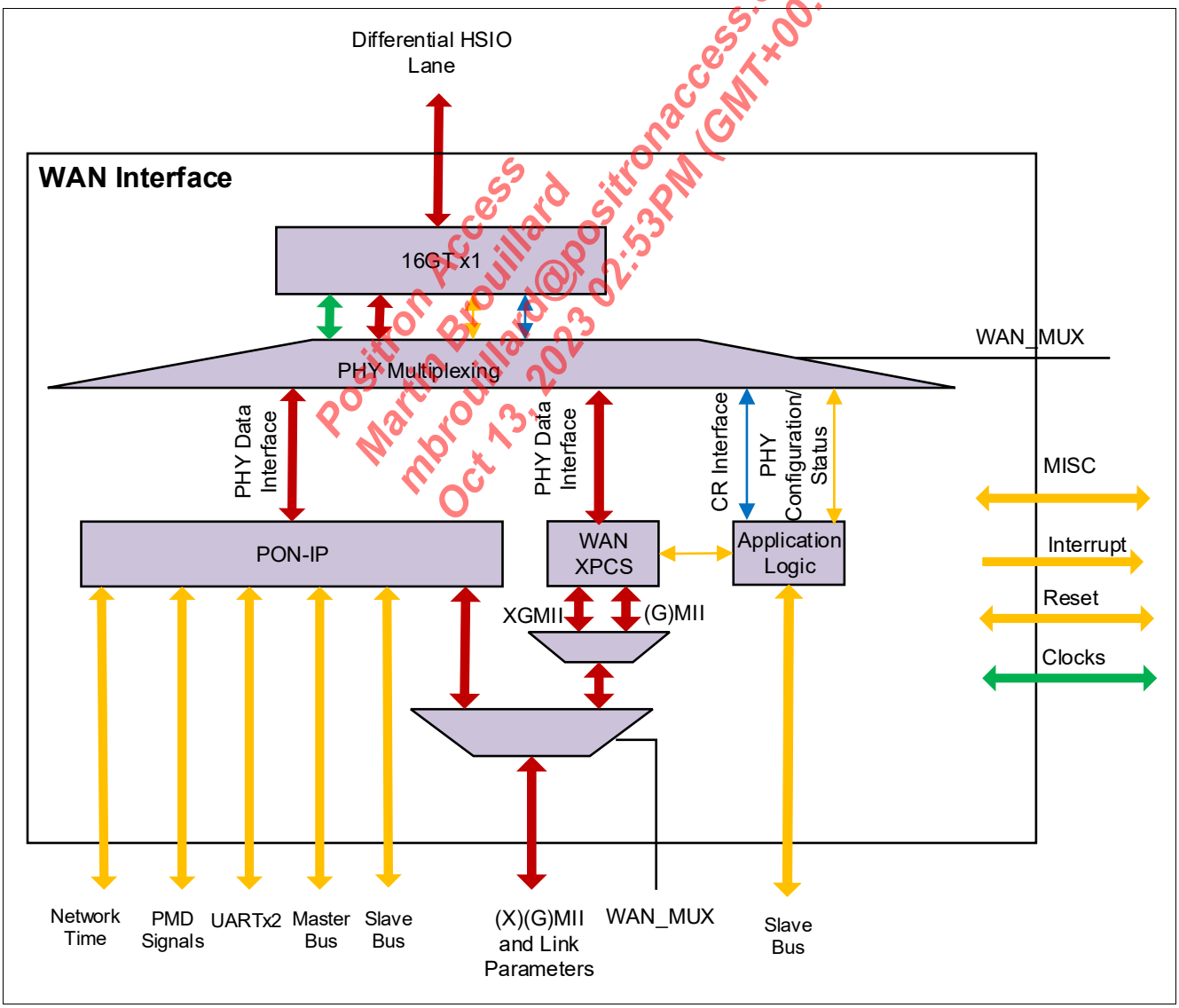


Figure 37 PON/XFI WAN Interface Module for URX851 and MxL25641

Figure 38 shows the block diagram of PON/XFI WAN interface module in URX850.

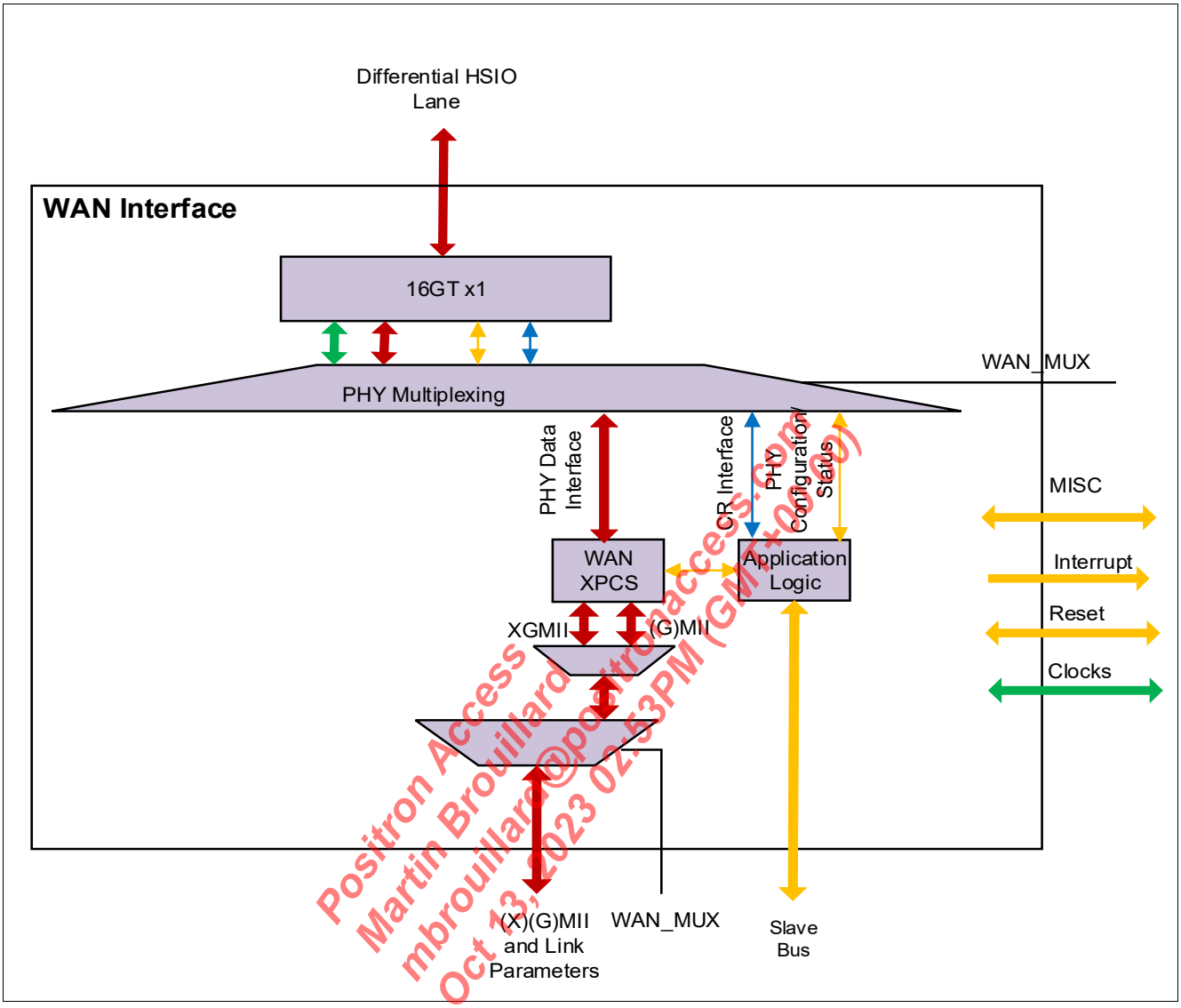


Figure 38 XFI WAN Interface Module for URX850

Attention: The WAN_MUX in Figure 38 is fixed to the Ethernet mode for URX850.

7.3.1.2 Features

This section provides the WAN interface module main features.

Applications

- Passive Optical Network (PON) mode
- Active Ethernet mode

External Interfaces

- SerDes analog I/O signals
  - A differential data input pair in receive direction
  - A differential data output pair in transmit direction
  - An external tuning resistor pin

Interrupt

- High active level interrupt outputs
  - Three from PON-IP
  - One from Ethernet XPCS
  - One from SerDes

High Speed SerDes

- Supports these PON high speed serial interfaces:
  - G-PON
  - XGPON-1
  - NG-PON2
  - XGSPON
  - EPON
  - 10G-EPON
  - Turbo EPON
  - DPoE (revised 10G-EPON)
- Supports these Ethernet high speed serial interfaces:
  - 1000BASE-(K)X (802.3ap) and 1 Gbps SGMII [8]: 1.25 GT
  - 10GBASE-(K)R: 10.3125 GT
  - 5GBASE-(K)R: 5.15625 GT
  - 2.5GBASE-(K)X and 2.5 Gbps SGMII: 3.125 GT
  - USXGMII Single port: 2.578125/5.15625/10.3125 GT
- Supports programmable internal reference clock selection
  - 156.25 MHz reference clock in active Ethernet mode
  - Loop from Rx CDR to Tx in PON mode
- Supports termination resistance tuning and sharing the reference resistor between multiple PHYs (with PHY at LAN side)
- Supports both transmit and receive equalization and adaption
- Supports Tx amplitude control
- Supports Rx loss of signal detection
- Supports independent Rx and Tx power state controls
- Supports Configuration Register (CR) port (control/configuration from application logic module)
- Test and Debug Support:
  - Scan test (at speed and stuck at)
  - Burn-in test
  - IDDQ test

  - Supports JTAG interface for PHY test (cascading JTAG interfaces of the multiple module are done in the top level)
  - Supports diagnostics for characterization and ATE testing: loopback, test pattern generation, BERT, error insertion and detection

Ethernet XPCS

- 1 instance of Ethernet PCS
  - Supports 8B/10B encoding (Clause 36 based)
  - Supports 64B/66B encoding (Clause 49 based)
- Supports Clause 73 auto-negotiation in backplane Ethernet mode
  - 1000BASE-(K)X, 1 Gbps, 1.25 GT
  - 10GBASE-(K)R, 10 Gbps, 10.3125 GT
  - 5GBASE-(K)R, 5 Gbps, 5.15625 GT
  - 2.5GBASE-(K)X, 2.5 Gbps, 3.125 GT
- Supports SGMII with Clause 37
  - 1 port, 1.25 GT, 8B/10B encoding
  - Supports auto-negotiation with these modes:
    1 Gbps, GMII mode
    100 Mbps, GMII mode
    10 Mbps, GMII mode
- Supports 2.5 Gbps SGMII
  - 1 port, 3.125 GT, 8B/10B encoding
  - 2.5 Gbps, GMII mode
- Supports 10G-SXGMII with Clause 37
  - 1 port, 10.3125 GT, 64B/66B encoding
  - Supports auto-negotiation with these modes:
    10 Gbps, XGMII mode
    5 Gbps, XGMII mode
    2.5 Gbps, XGMII mode
    1 Gbps, GMII mode
    100 Mbps, GMII/MII mode
    10 Mbps, GMII/MII mode
- Supports 5G-SXGMII Clause 37 per Ethernet PCS
  - 1 port, 5.15625 GT, 64B/66B encoding
  - Supports auto-negotiation with these modes:
    5 Gbps, XGMII
    2.5 Gbps, XGMII
    1 Gbps, GMII
    100 Mbps, GMII/MII
    10 Mbps, GMII/MII
- Supports 2.5G-SXGMII Clause 37 per Ethernet PCS
  - 1 port, 2.578125 GT, 64B/66B encoding
  - Supports auto-negotiation with these modes:
    2.5 Gbps, XGMII
    1 Gbps, GMII
    100 Mbps, GMII/MII
    10 Mbps, GMII/MII
- Clause 37 at MAC side only
- Supports Clause 72 for 10G-BASE (K)R training
- Supports forward error correction in 10G-BASE (K)R mode
- Supports EEE mode

- Supports APB for configuration

PON-IP

- Supports G-PON, XGPON-1, XGSPON, and NG-PON2 ITU standards.
- Supports EPON and 10G-EPON IEEE standards.
- Also supports Turbo EPON (non-standard double rate EPON) and DPoE (modified 10G-EPON) modes.
- Plain Ethernet modes not supported
- Supports 64 LLIDs
- MPCP handling in US and DS
- Control character handling (idle word deletion/insertion, special tag insertion, timestamp append) in US and DS
- OMCI / MPLS packet encapsulation for XGMII interface
- Reassembly buffer
- XGMII interface
- XGMII flow control
- 8B/10B and 64B/66B PCS layers
- FEC in upstream and downstream direction
- AES en-/decryption as per DPoE-SP-SEC and GPON standards
- Triple churning en-/decryption as per CTC requirement
- Supports 256 XGEM ports
- Ethernet filtering
- DBRu reporting modes 0 and 1
- Preemption support in downstream direction
- Multiple PLOAMu messages per burst
- ToD stamping for 1588 and OAM
- Synchronization of/by external ToD counter (sync-in/sync-out)
- RX-LOS indication
- Configurable interface to SerDes
- Mailbox interface for firmware/software communication
- OMCI firmware interface
- IO-Bus master interface for Tx data request and backlog information
- ToD interface
- UART debug interfaces
- DFT ports
- ARC powered micro-controller subsystem
- Flexible firmware driver support, with alarm and status reporting
- Linux based software control with performance monitoring
- Management software control based on managed objects
- Supports different hardware/software power save modes

Application Logic

- One instance of application logic module
- APB bridge to PHY control register interface
- APB bridge to Ethernet XPCS APB configuration interface
- Supports PHY configuration and status

Multiplexing and glue Logic

- Multiplexing data path between PON-IP, Ethernet XPCS, and PHY
- Multiplexing XGMII and (G)MII interface of XPCS
- Multiplexing clock, control, and status signals between PON-IP, Ethernet PCS, and PHY
- Multiplexing XGMII interface and link parameters from PON-IP and XPCS to GSWIP-O subsystem

7.3.1.3 Hardware Description

This section provides a detailed functional description of the subsystem.

Ethernet PCS Configuration

Table 93 summarizes all the supported modes.

XPCS only works at the MAC side.

The mode is statically set by the configuration.

The submode is determined by auto-negotiation Clause 73 and the speed is determined by auto-negotiation Clause 37.

The EEE mode is supported in all modes.

Clause 72 and FEC are supported for 10G-KR mode.

For each link, there are a separate XGMII interface and GMII/MII interface from the XPCS. A multiplexer is required to mux these two interfaces according to the selected modes.

Table 93 XPCS Support Modes

Modes	Sub-mode Auto-negotiation	Submodes	Baud Rate	Coding	Link Speed Auto-negotiation	Link Speed	XGMII /GMII/MII Modes
Backplane Ethernet	Clause 73	10G-(K)R	10.3125 GT	64b/66b	No	10 Gbps	XGMII
		5G-(K)R	5.15625 GT	64b/66b	No	5 Gbps	XGMII
		2.5G-(K)X	3.125 GT	8b/10b	No	2.5 Gbps	XGMII
		1000-(K)X	1.25 GT	8b/10b	No	1 Gbps	GMII
SGMII-1G	No	SGMII-1G	1.25 GT	8b/10b	Clause 37	1 Gbps, 100 Mbps, 10 Mbps	GMII, GMII/MII, GMII/MII
SGMII-2.5G	No	SGMII-2.5G	3.125 GT	8b/10b	No	2.5 Gbps	GMII
10G-SXGMII (1 Port)	No	10G-SXGMII	10.3125 GT	64b/66b	Clause 37	10 Gbps, 5 Gbps, 2.5 Gbps, 1 Gbps, 100 Mbps, 10 Mbps	XGMII, XGMII, XGMII, GMII, GMII/MII, GMII/MII
5G-SXGMII (1 Port)	No	5G-SXGMII	5.15625 GT	64b/66b	Clause 37	5 Gbps, 2.5 Gbps, 1 Gbps, 100 Mbps, 10 Mbps	XGMII, XGMII, GMII, GMII/MII, GMII/MII
2.5G-SXGMII (1 Port)	No	2.5G-SXGMII	2.578125 GT	64b/66b	Clause 37	2.5 Gbps, 1 Gbps, 100 Mbps, 10 Mbps	XGMII, GMII, GMII/MII, GMII/MII

7.3.2 GSWIP-O Module

The GSWIP-O in the URX851/URX850/MxL25641 is tailored for bridging the L2 data plane and QoS functions as defined in GPON (G.988) and SIEPON (P1904.1). The subsystem allows straight-forwarding mapping from the Linux bridge and closely follows the G.988 OMCI packet processing model and BBF TR-156 specification.

Table 94 shows the maximum number of supported ports/interfaces in the GSWIP-O. The number of ports/interfaces varies depending upon the device.

Table 94 Number of Ports/Interfaces per Product

	Number of Ports/Interfaces per Product		
Component	**URX851**	**URX850**	**MxL25641**
10G-MAC and Legacy GMAC	9	9	7
10G-MACsec	1	1	1
MDIO Master	2	2	2

The GSWIP-O includes these common components across all devices:

- One instance of GSWIP3.2 core (configured for GSWIP-O)
- Memory for the GSWIP-O core
- Multiplexer and glue logic

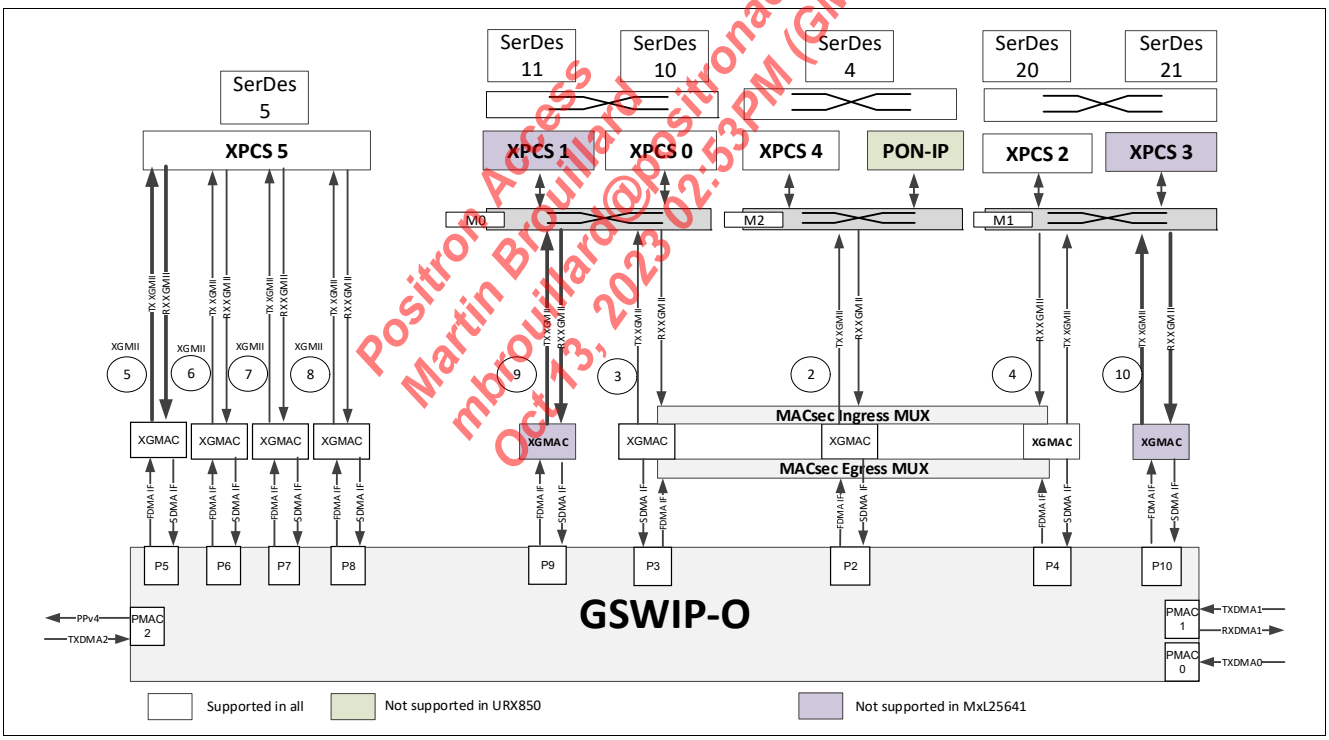


Figure 39 Streaming Port Muxing

7.3.2.1 Overview

Figure 40 shows the block diagram of GSWIP-O module in the URX851, URX850, and MxL25641 devices.

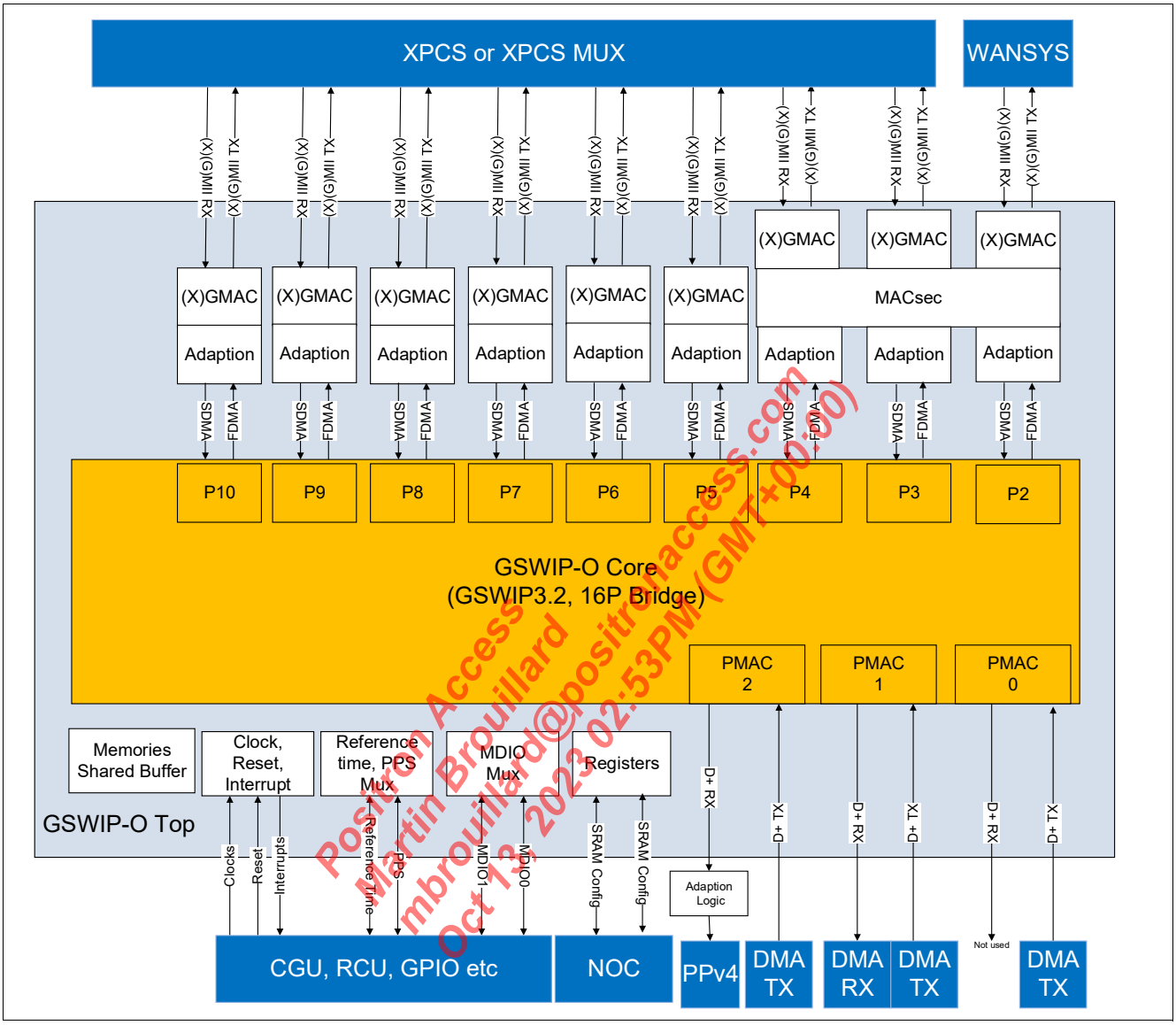


Figure 40 GSWIP-O Module

7.3.2.2 Features Description

This section provides an overview of the GSWIP-O module main features.

10G-MAC and Legacy MAC

- The nine instances of Ethernet MAC (including10G MAC and Legacy GMAC each) are compliant with IEEE 802.3 and support these features:
  - The XGMII interfaces comply with IEEE 802.3 Clause 46
  - The GMII interfaces is comply with IEEE 802.3 Clause 35
  - The MII interface comply with IEEE 802.3 Clause 22
  - 10/100/1000/2500/5000/10000 Mbps standard operation speed
  - Flexible operation speed
  - Full-duplex operation mode (for all speeds) and half-duplex operation mode (only for 10/100 Mbps standard operation speed)
  - Enhanced frame size support (“Jumbo frames”, programmable limit up to 10 Kbyte)
  - Pause frame and 802.1Qbb PFC pause frame in full duplex mode
  - Backpressure (forced collisions) in half-duplex mode
  - Frame padding on egress traffic
  - Minimum frame length check and maximum frame length check. Maximum frame length are configurable.
  - FCS verification and stripping
  - FCS generation and insertion
  - Supports Low Power Idle (LPI) mode as defined by IEEE 802.3az

G.INT

- Supports G.999.1 native encapsulation (with length field and without length field) and Ethernet encapsulation
- Supports multiple streams (up to 32) and the number of streams per interface is configurable (4, 8, 16)
- The stream ID first valid bit position is configurable.
- Supports G.INT flow control in both transmit and receive direction

MACsec

- MACsec which is configurable and attachable to one of the three MACs (MAC2, MAC3 and MAC4)
- Compliant with IEEE 802.1AE. Supports 128-bit and 256-bit encryption, 16 security channels, and 32 security associations in each direction
- Supports classification based on the special tag and other packet header fields
- Supports bypass for non-MACsec packets
- Supports transparent forwarding based on certain fields in the packet. These fields are the special tag after the source MAC address and the receive time stamp at the end of the packet.
- Supports the special tag packet size field update for encrypted packets in the transmit direction

1588 Time Stamp

- Supports Ethernet packet time stamping as IEEE 1588 (v1 and v2)
- Supports ordinary clock, boundary clock, end to end transparent clock, and peer to peer transparent clock modes
- Supports PTP over Ethernet and PTP over UDP when using IPv4/IPv6.
- Supports both external reference time (from PON and other Ethernet interfaces) and internal reference time. The internal reference time and frequency is adjustable.
- Supports auxiliary time stamp snapshot with two external events
- Supports 1 PPS (pulse per second) output
  - Supports both fixed and flexible (programmable for start, stop, pulse width, interval) PPS output.
- Supports time stamp snapshotting of the packet at the start frame detection (SFD) in both directions. Supports programmable ingress time stamp correction and programmable egress time stamp correction.

- Attaches a snapshot time stamp to end of the every received packet
- Supports one step time stamping (64 entries FIFO per Ethernet interface) in the Tx direction
- Supports two step time stamp (16-entries in FIFO per Ethernet interface) in the Tx direction

MDIO Master

- Contains two MDIO master interfaces which allow control of external MDIO-configured devices.
- Supports both Clause 22 and Clause 45.
- Automatically scans the link status change for the 9 external devices. The MDIO addresses of the external devices are programmable.
- Supports direct link parameters input (speed, duplex, flow control, and link status) from XPCS
- The management agent is able to access the external devices via MDIO interface.

GSWIP-O Core Processing

- PMAC
  - Supported Egress functions include: FCS replacement or FCS insertion
  - Supported Ingress functions include: FCS verification and stripping
  - Frame padding on egress traffic
  - Checks the minimum and maximum frame length. The frame lengths are configurable.
  - Backpressure signals to DMA to regulate the traffic from DMA to GSWIP.
  - IPv4 checksum and TCP/UDP checksum verification, checksum error indication and counter on the egress traffic
  - IPv4 checksum and TCP/UDP checksum re-generation on the ingress traffic.
- Processing path
  - Header-only mode for memory port traffic
  - PCE bypass mode for traffic from PMAC and MAC
  - Wake-on-LAN
  - Packet statistics
  - Packet mirroring for classified traffic and PCE-bypass traffic
- Packet Classification
  - Supports sub-interface classification (for example LLID, GEM ID, and VLAN ID) over the single physical interface
  - L2 bridging at sub-interface level
  - L3 Multicast processing at sub-interface level
  - PBB (MACinMAC)
  - VLAN filtering
  - Extended VLAN tagging operation
  - Traffic flow classification
  - Packet filtering
  - Port trunking
  - Packet insertion and extraction
  - Ethernet OAM delay and loss measurement support
  - Ethernet OAM loopback support
  - Packet loopback
- QoS
  - Traffic QoS classification
  - Priority and color marking/remarking
  - Traffic metering
  - Pre-parser priority classification based on VLAN priority bits and DSCP field.
  - Pre-parser priority based ingress congestion control (802.1Qbb flow control and selective earlier discard)

7.3.2.3 Hardware Description

This section provides a detailed functional description of the GSWIP-O.

GSWIP-O is responsible for classifying, storing, and forwarding of multiple data flows. The switch consists of storage buffer, packet queuing, and packet classification units. Ingress data is received on one of the port interfaces, classified, and placed in the appropriate QoS queue in the shared buffer. Ingress policing and access control rules are applied to the received traffic and packets which do not comply with the the rules are discarded. Prior to a packet being fetched from the shared memory and transmitted on one of the egress ports, it is subject to egress scheduling, rate shaping, and modification.

7.3.2.3.1 MDIO Master Interface

The Management Data Input/Output (MDIO) master interface provides the register interface access to external or internal PHY registers, external switch registers, and automatic PHY status scanning function.

There are two MDIO master interfaces. The MDIO master interface is controlled by logic inside XGMAC2 and XGMAC3.

The MDIO master interface inside XGMAC performs these tasks:

- Supports MDIO Clause 22 and Clause 45 frame structure.
- Allows programming the MDC clock
- Provides the management agent MDIO access for these activities:
  - The management agent accesses external PHY/switch or internal PHY through MDIO access. Each MDIO access includes the start type (Clause 22 or Clause 45), the access type (address/read/write/post-increment command), the 5-bit PHY address, the 5-bit register/device address, and the 16-bit read/write data.
  - When the MDIO Master interface is accessed by the management action, through indirect Master MDIO access registers, it has a higher priority over the auto-scanning state machine.
- Auto-scanning process details:
  - This command helps reduce the load on the software for checking the link ok status of the devices on each port. This command has the lowest priority and can be enabled to always run in the background.
  - Auto-scanning for each link can be independently enabled.
  - The Clause 22 or Clause 45 mode is programmable for each link.
  - Up to 9 links are supported for auto-scanning.
  - The MDIO address for each link is programmable.

MDIO Access Description

The MDIO interface uses the serial protocol defined by IEEE 802.3, Clause 22 or Clause 45.

High Speed Operation

The standard MDIO protocol uses a clock rate of 2.5 MHz on the MDC. This is the default setting. To speed up the data exchange, it is possible to increase the clock generated on the MDC, when the connected PHY is able to support it.

Data Protocol

The MDIO is operated in master mode only. It provides an indirect access to the PHY registers via the MDIO serial interface.

Figure 41 and **Figure 42** show the serial interface protocol.

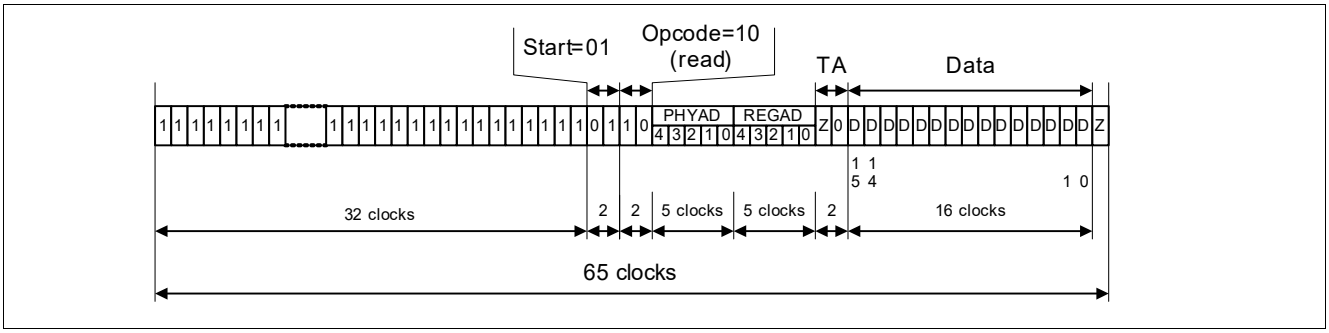


Figure 41 Clause 22 MDIO Read Access

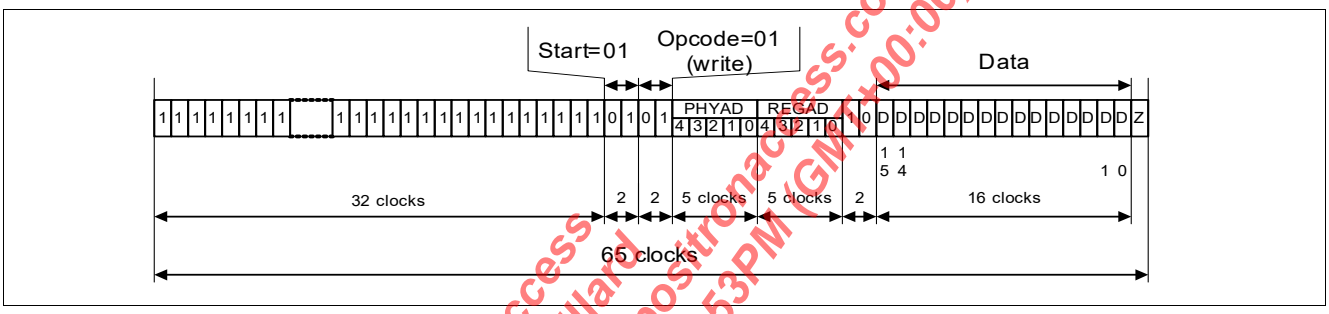


Figure 42 Clause 22 MDIO Write Access

Table 95 Clause 22 Access Format

Frame	Description
IDLE	The MDIO line is three-state. There is no clock on `xgmii_mdc_o`.
PREAMBLE	32 bits of continuous ones
START	Start of frame is 2'01
OPCODE	• 2'b01 for write • 2'b10 for read
PHY ADDR	5-bit address select for one of 32 PHYs
REG ADDR	5-bit register address to select the register within each MMD
TA	Turnaround • 2'bZ0 for read • 2'b10 for write where Z is the tri-state level
DATA	Any 16-bit value • In a write operation, the XGMAC drives MDIO. • In a read operation, the PHY drives MDIO.

Table 96 Clause 45 Access Format

Frame	Description
IDLE	The MDIO line is three-state. There is no clock on `sma_mdc_o`.
PREAMBLE	32 bits of continuous ones
START	Start of frame is 2'00
OPCODE	• 2'b00 for address • 2'b01 for write • 2'b10 for post-read increment address • 2'b11 for read
PHY ADDR	5-bit address select for one of 32 PHYs
DEV ADDR	5-bit address select for one of 32 devices
TA	Turnaround • 2'bZ0 for read and post-read increment address • 2'b10 for write and the address accessed by the MDIO where Z is the tri-state level
DATA/ADDRESS	16-bit value. For an address cycle (OPCODE = 2'b00), this frame contains the address of the register to be accessed on the next cycle. For the data cycle of a write frame, this field contains the data to be written to the register. For read or post-read increment address frames, this field contains the contents of the register read from the PHY. • In address and data write cycles, the XGMAC drives the MDIO line during the transfer of these 16 bits. • In read and post-read increment address cycles, the PHY drives the MDIO line during the transfer of these 16 bits.

7.4 Ethernet PHY

This section describes the 2.5 Gbps Ethernet PHY:

- **Overview of the Ethernet PHY Module** (**Section 7.4.1**)
- **Feature List Of Ethernet PHY** (**Section 7.4.2**)

Attention: MxL25641 does not support embedded Ethernet PHYs

7.4.1 Overview of the Ethernet PHY Module

The Ethernet PHY module is a quad port Ethernet PHY with each port providing a copper interface to CAT5e or better cables supporting 2.5GBASE-T(IEEE802.3bz), 1000BASE-T(IEEE802.3 Clause 40), 100BASE-Tx (IEEE 802.3 Clause 25), and 10BASE-Te (IEEE 802.3 Clause 14).

Using the terminology defined by the Open System Interconnect (OSI) model, this product implements a layer 1 physical media access device.

Each port is connected to a layer 2 MAC using a GMII interface and an MDIO management interface.

An external supply of 0.9 V, 1.8 V, and 3.3 V is required.

Figure 43 shows an overview of the URX851/URX850's Ethernet PHY module.

*Note: Not all PHYs on all devices support 2.5GBASE-T. See **Table 4** for more information.*

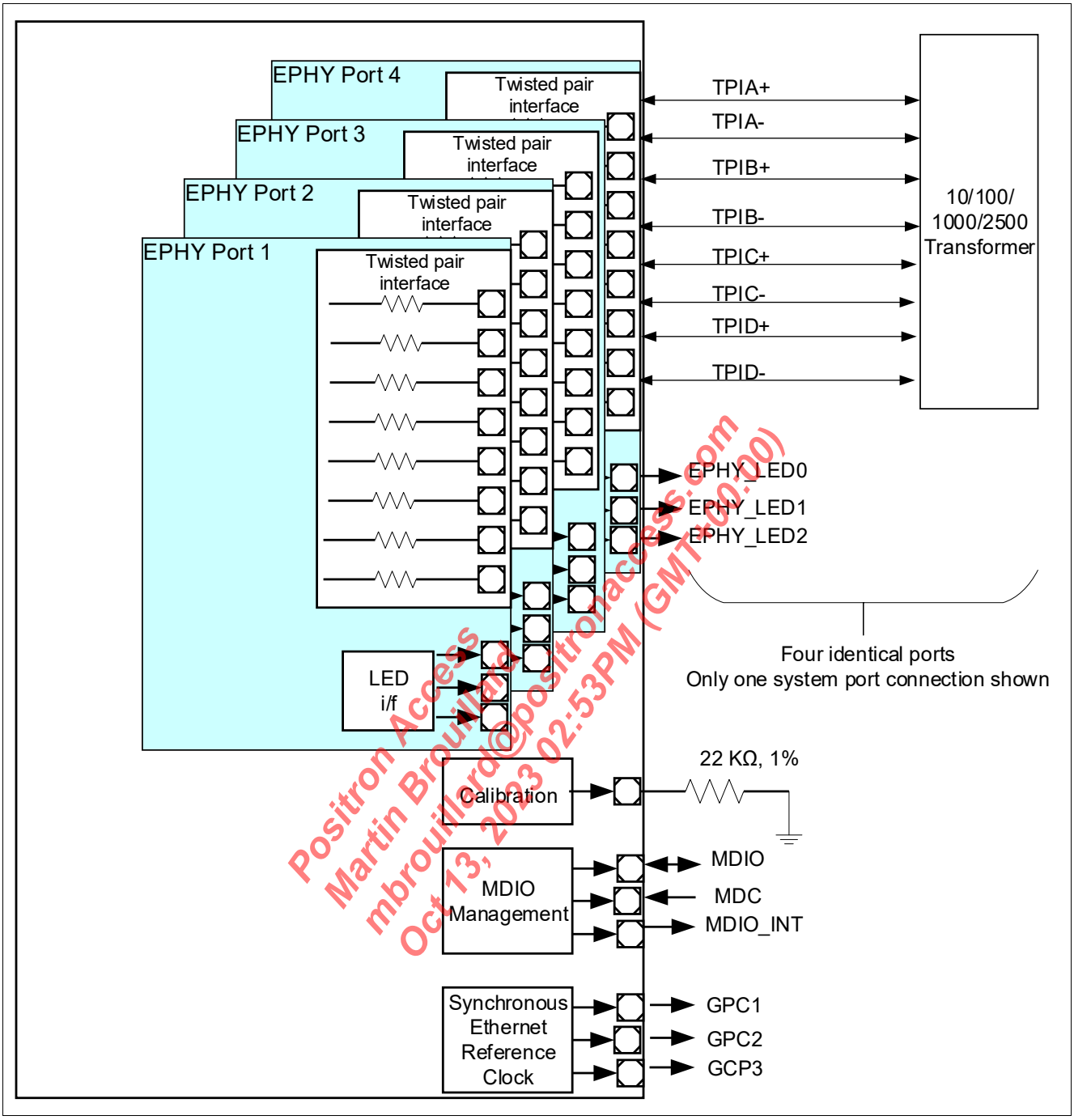


Figure 43 URX851/URX850 Ethernet Interface Block Diagram

7.4.2 Feature List Of Ethernet PHY

This section provides the Ethernet PHY features.

Interfaces

- Four multiple speed Ethernet PHY interfaces, compliant with:
  - 10BASE-Te
  - 100BASE-TX
  - 1000BASE-T
  - 2.5GBASE-T (IEEE802.3bz, NBASE-T)
  - Auto-MDIX
  - Auto-downspeed
  - Auto-negotiation with extended next page support
  - Cable diagnostics: cable open/short detection and cable length estimation
  - Test loops and analog self test
  - Power down modes
  - SyncE according to ITU-T G.8262/Y.1362
  - Supports transformer-less Ethernet for backplane applications
- MDIO slave interface to allow control from an external master, such as router or microcontroller:
  - Supports Clause 22 and Clause 45
- JTAG boundary scan, test, and debug interface
  - Share pins with GPIO functions
  - UART interface (share pins with GPIO)
- Status indicating LEDs:
  - Directly attached
  - Up to 3 LEDs per port
  - Configurable LED functions per LED (link/activity, duplex/collision, link speed etc)
  - Steady/blinking indication
- GPIO Pins shared with general purpose interrupt, general purpose clock, MDIO slave, LED, JTAG, and debug functions
- Supports two external interrupts:
  - Configurable as output to an external controller
  - Configurable as input from external device(s)
  - Configurable edge, level, and polarity

Clocking

- Supports various reference clock speeds:
  - Direct clock input mode with 25 MHz
- An integrated PLL in central distribution block (CDB) generates the GPHY IP and module system clocks.
- Supports three external clock pins:
  - Configurable as output to external device(s)
  - Configurable as input from external device(s): for example, used for SyncE reference clock
  - Multiple clock speed: 25 MHz, 2.048 MHz, 1.544 MHz, or 8 kHz. Share pins with GPIO functions
  - Input or output signal-based time stamping allows timer synchronization with external devices

Other Features

- 802.3az energy efficient Ethernet
- IEEE 1588v2: one-step and two-step time stamping
- SyncE
- Temperature sensor (warning, interrupt, and auto-downspeed)

7.5 USB Subsystem

The USB subsystem description covers these sections:

- **Overview** (**Section 7.5.1**)
- **USB 3.2 Subsystem Features** (**Section 7.5.2**)
- **USB 3.2 Host Controller** (**Section 7.5.3**)
- **USB 3.2 Gen 1 and Gen 2 Transceiver** (**Section 7.5.4**)
- **Features** (**Section 7.5.5**)
- **USB Subsystem External Interfaces** (**Section 7.5.7**)

The USB subsystem provides USB 3.2 Gen 2 x1 host interfaces:

- URX851/URX850: 2x interfaces port with USB Type-C ports
- MxL25641: 1x interface port with USB Type-A port

USB 3.2 Gen 2 x1 ports support 10 Gbps/5 Gbps/480 Mbps/12 Mbps transfer speeds.

The USB subsystem supports these applications:

- USB 3.2 SuperSpeedPlus (SSP)/SuperSpeed host with backward compatible to USB 2.0 high speed
- USB 3.2 SSP/SuperSpeed host only
- USB 2.0 high speed host only
- Concurrent USB 3.2 SSP/SuperSpeed and USB 2.0 high speed
- DFU Boot with controller set to device mode

The Type-C features are:

- Integrates two USB 3.2 Gen 2 SerDes lanes
- Integrates Type-C assist block which does the lane switching based on Type-C Port Manager software
- Integrates a USB Type-C assist block which provides lane switching based on managerial software via an external CC controller. Supports CC1 and CC2 pins for cable attachment and removal detection, as well as cable orientation detection.
- Supports both USB 3.2 and USB 2.0 connections
- Does not support USB Battery Charging v1.2 Specification on a USB 2.0 lane.
- Supports the USB Power Delivery 2.0 protocol over CC via an external controller.
- Supports alternate third-party modes via an external controller.

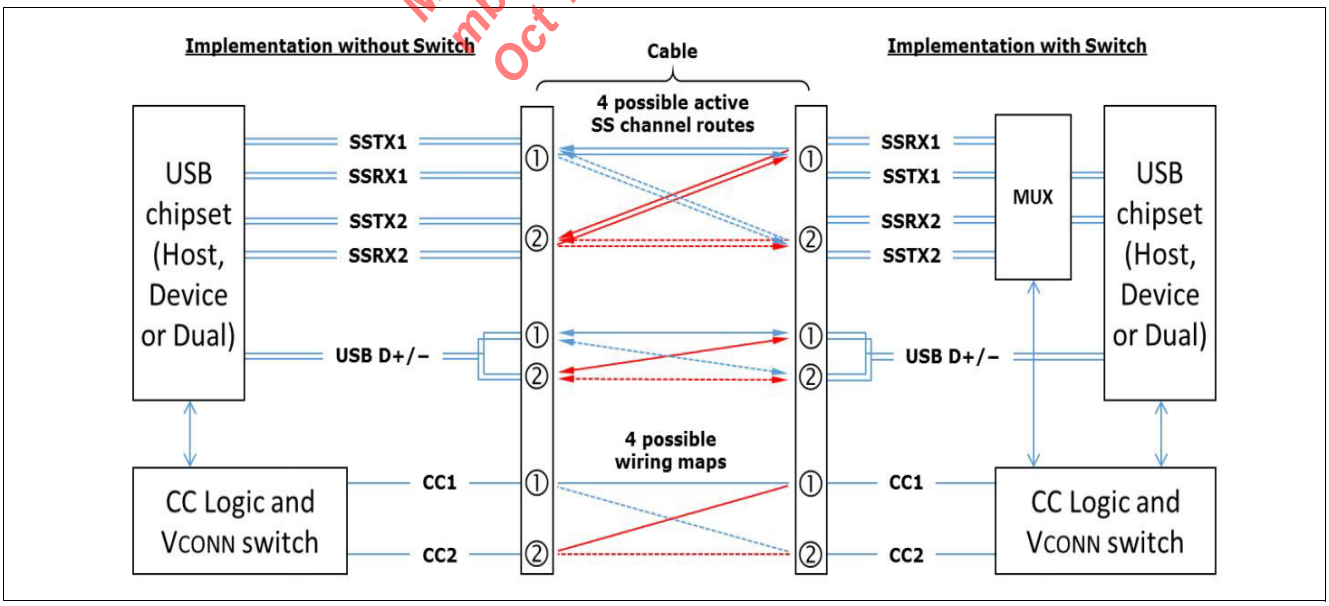


Figure 44 Type-C Connections (Only Single CC Is Wired Inside the Type-C Cable)

The Type-A features are:

- Integrates single USB 3.2 Gen 2 x1 SerDes lane
- Supports both USB 3.2 and USB 2.0 connections
- Does not support USB Battery Charging v1.2 Specification on a USB 2.0 lane.

The USB subsystem includes these blocks:

- USB 3.2 Host Controllers
- USB 3.x Transceivers
- USB 2.0 Transceivers
- System interface (AXI Bus slave and DMA including AXI Bus master)

Attention: The USB Implementers Forum updated the USB specification for the third generation USB from USB 3.1 to USB 3.2. Refer to Table 3 for the interface name mapping.

7.5.1 Overview

The USB 2.0/3.2 host interface is integrated according to **Figure 45** to **Figure 46**.

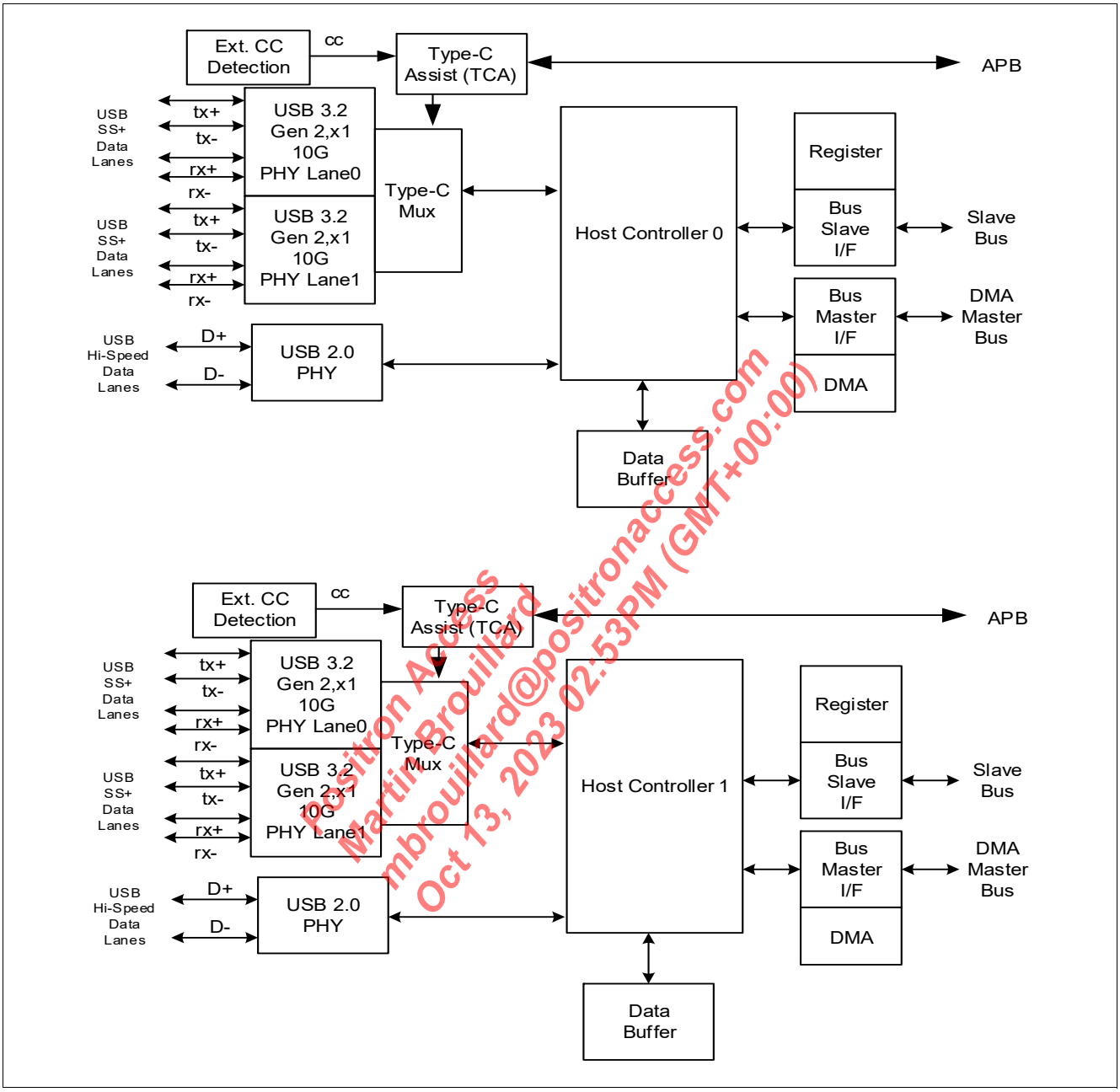


Figure 45 URX851/URX850 USB 3.2 Gen 2 Subsystem Block Diagram

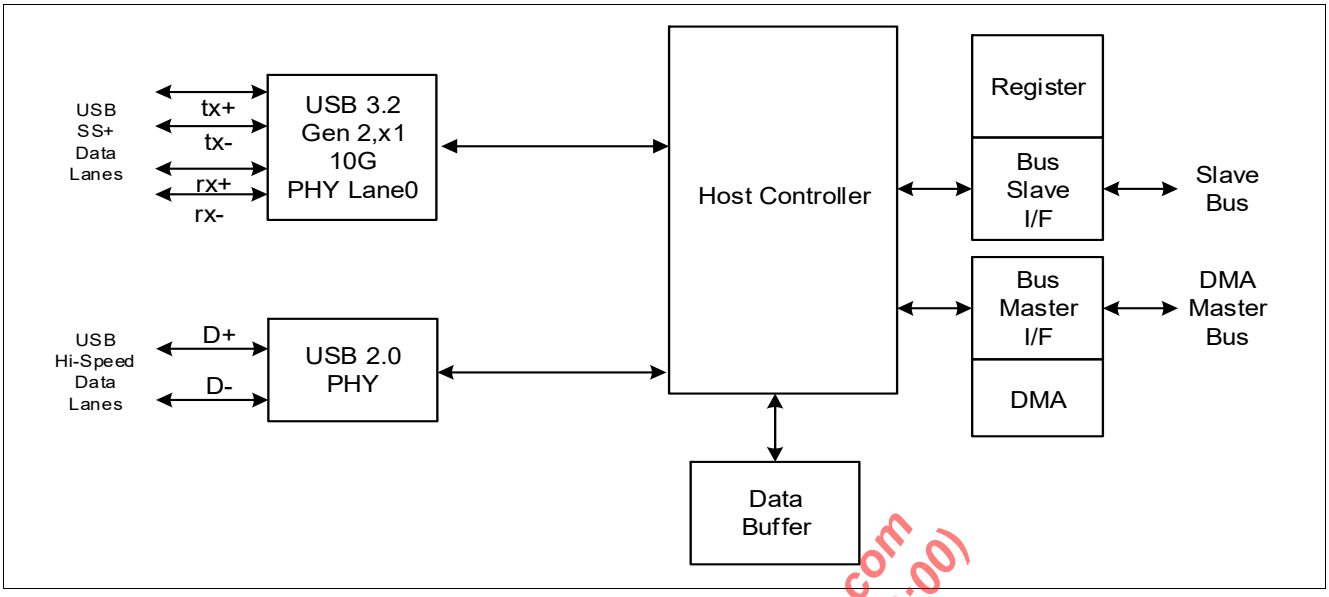


Figure 46 MxL25641 USB 3.2 Gen 2 Subsystem Block Diagram

7.5.2 USB 3.2 Subsystem Features

The major features of USB subsystem are:

- USB 3.2 compliance
- Gen 2 SSP (10 Gbit/s) host
- Gen 1 Super speed (5.0 Gbit/s) host
- High speed (480 Mbit/s), full speed (12 Mbit/s) operation for host
- Supports control, bulk, interrupt and isochronous USB transfers (host)
- Streaming mode, 1 to 16 bursts
- Built-in root hub (Host)
- Concurrent read and write operations for USB 3.2 full duplex mode
- Concurrent USB 3.2/2.0 traffic
- xHCI 1.1 compatible
- Descriptor caching to prevent system latency variation
- Protocol-ware DMA engine, configurable for buffer DMA mode and descriptor based DMA mode
- Advanced latency buffer architecture to minimize FIFO requirements
- Programmable host channels up to 16
- When running in host mode, the controller supports a maximum of 12 active asynchronous and 12 active periodic endpoints concurrently
- Hardware controller LPM according to USB 2.0 Link Power Management Addendum (LPM-ECN)
- Automatic clock gating and switching to suspend clock for MACs
- Integrated in USB 3.2/2.0 high speed transceiver
- Overcurrent protection

Application Notes for Isochronous Support

- The embedded USB ports support USB isochronous applications such as webcams.
- MaxLinear recommends directly connecting isochronous devices to the host root port without any hub if possible.
- MaxLinear recommends not directly connecting any FS isochronous device behind a HS hub if using a hub cannot be avoided.

7.5.3 USB 3.2 Host Controller

USB 3.2 is a high performance bus. In Gen 2 mode, the raw link speed is 10 Gbps while in Gen 1 mode, the raw link is 5 Gbps. In other words, a 1 KB of packets takes only 2.65 µs on the cable, while 64 KB packets takes only 164 µs with newly burst transfers. This level of performance requires architecture and system connection support to provide low latency, a wide bus, and high frequency operation.

The USB host controller writes to a target device by reading data from memory supplied by the USB device driver, then it performs a parallel to serial conversion, which creates a USB transaction. This data is forwarded to the USB interface. For read transfers, the host sends out the read transaction to the target device. The target device recognizes that it is being addressed and that it's data is being requested. The target device then transfers the requested data to the host. The USB host controller performs serial to parallel conversion and transfers the data to the device driver’s memory buffer.

In addition to data transfer, the USB host control also enables or disables USB ports, controls the power supply to ports, attaches and removes devices from the system, and sets and reports status reports to the software.

System Bus Interface

The system bus master interface is used for transferring data between the system's core RAM and system memory using DMA.

The bus slave interface provides software access to internal registers. It also provides debug access to all internal RAM.

Control and Status Register

The Control and Status Registers control the internal hardware functional modules.

List Processor

The List Process interprets the USB transfer descriptors setup using software and manages USB transfers. The List Process schedules transfers for multiple USB devices.

Buffer Management Unit

The buffer management unit manages descriptors and data fetches.

U3/U2 Protocol Transaction Layer

The Protocol Transaction Layer handles events such as toggle/retry for USB 2.0 and bursts for USB 3.2.

U3/U2 MAC

The U3/U2 MAC controls packet Tx/Rx, the downstream port state machine, CRC, and link management.

U3 Link

The U3 Link controls packet framing, checking, skip and link training, control over link, and power state.

7.5.4 USB 3.2 Gen 1 and Gen 2 Transceiver

Inside the document, USB 3.2 Gen 2 may also be called USB 3.1; USB 3.2 Gen 1 may be also called USB 3.0.

7.5.4.1 Features

The USB PHY provides these features:

- Compliant with the Universal Serial Bus Specification, Revision 3.2
- Optimized for low power dissipation while active, idle, or on standby
- No external components are required, but for best operation only a single resistor is required.
- Provides on-chip low-jitter PLL to reduce clock noise
- Supports 10/5 Gbps high-speed data transmission rates through a 3 meter USB 3.x cable.
- SSC generation and absorption
- Supports 32-bit interface
- Implements logic to support U0, U1, U2, and U3 power saving modes
- Supports USB 3.x test modes

7.5.5 USB 2.0 PHY

Figure 47 shows that the USB 2.0 PHY consists of two basic components, the common block and the transceiver block. The common block contains the Phase-Locked Loop (PLL), bias, and Crystal Oscillator (XO) blocks. The transceiver block comprises the analog block and the digital block, which control the transmit, receive, disconnect detection, squelch detection, and automatic tuning functions.

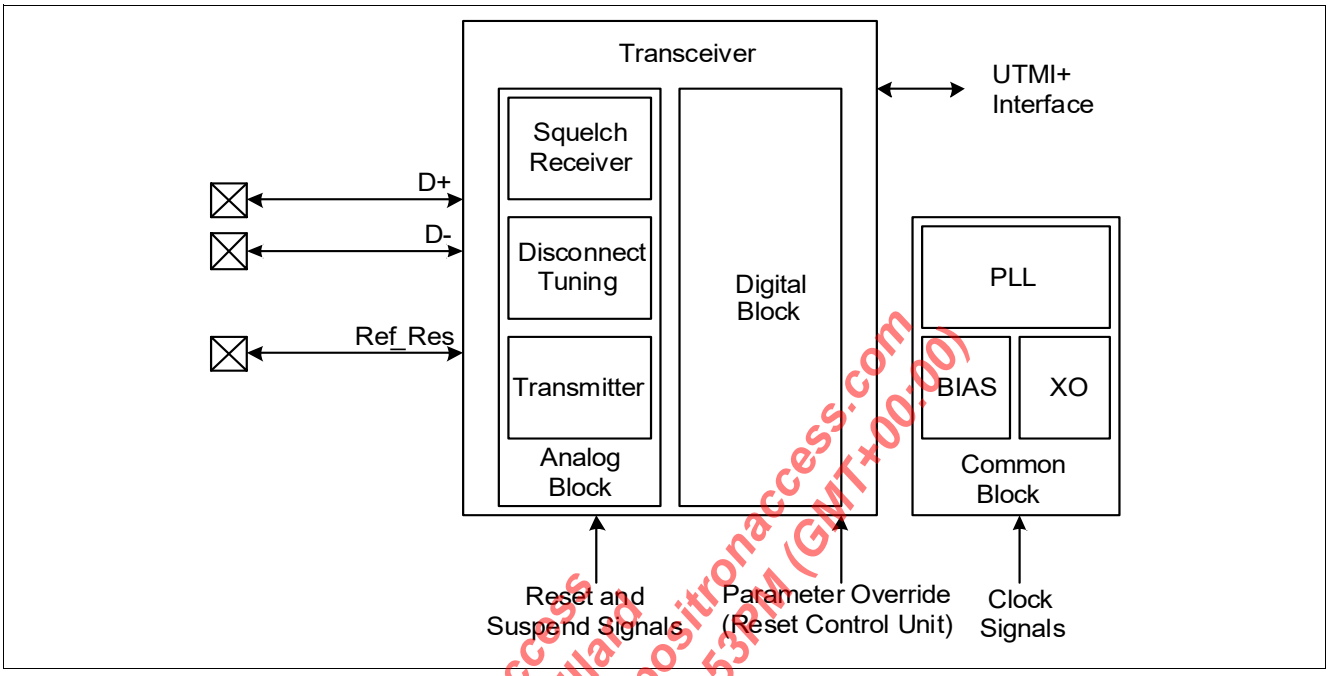


Figure 47 Overview of the USB PHY

7.5.5.1 Features

The USB 2.0 PHY provides these features:

- Compliant with the Universal Serial Bus Specification, Revision 2.0
- Optimized for low power dissipation while active, idle, or on standby
- Provides parameter override bits for optimal yield and interoperability
- No external components are required, but for best operation only a single resistor is required.
- Provides on-chip PLL to reduce clock noise
- Fully integrates short-to-5 V and short-to-ground protection for D+ and D- lines
- Fully integrates 45 Ω termination, 1.5 kΩ pull-up, and 15 kΩ pull-down resistors, with support for independent control of the pull-down resistors
- Supports 480 Mbps High-Speed (HS), 12 Mbps Full-Speed (FS), and 1.5 Mbps Low-Speed (LS) (Host mode only) data transmission rates
- Supports 16-bit unidirectional parallel interfaces for HS, FS, and LS (Host mode only) modes of operation, in accordance with the UTMI+ Specification, Revision 1.0 (Level 3 without OTG support)
- Provides dual (HS/FS) mode host support
- Implements data recovery from serial data on the USB connector
- Implements SYNC/EOP generation and checking
- Implements bit stuffing and unstuffing, and bit-stuffing error detection
- Implements Non-return-to-zero, inverted (NRZI) encoding and decoding
- Implements bit serialization and deserialization
- Implements holding registers for staging transmit and receive data
- Implements logic to support suspend, resume, and remote wakeup operations
- Supports USB 2.0 test modes

7.5.6 Type-C Assist Block

The URX851/URX850 solution for Type-C eliminates the need for an external Type-C switch component.

The solution is minimalistic in terms of power usage. It requires less power than two separate USB 3.1 PHYs due to shared common blocks and is cost effective because it does not require an external Type-C switch component.

The PHY must be in a known state before performing these actions:

- Switching lanes (changing cable orientation)
- Moving unused lanes to lower power states.

Performing these actions requires synchronization between `USB_Ctrl` and USB Type-C PHY. This is achieved with minimal changes to the conventional USB Type-C system design and is facilitated by the X-Bar Assist Block.

In addition to the MUX switching requirements, USB Type-C also requires manipulation of `DrvHost` VBUS functionality and timing to assert `VBUSValid` to the USB Subsystem.

These requirements are due to the fact that for Type-C connectors, VBUS is only switched on when a Type-C connection is detected. These and other VBUS requirements are ensured by the VBUS assist block.

7.5.7 USB Subsystem External Interfaces

The USB subsystem implements D+ and D- signals that directly interface to the USB connector. A USB transmitter resistor tune pin provides connectivity for an external resistor that adjusts the USB PHY's high-speed source impedance. **Figure 48** shows the external USB subsystem signals.

When a USB subsystem is configured for host mode, it also generates an overcurrent indicator as an interrupt request to the ICU. In a USB overcurrent condition the software turns off power to the USB power switch IC, thereby cutting off the power supply to downstream USB devices.

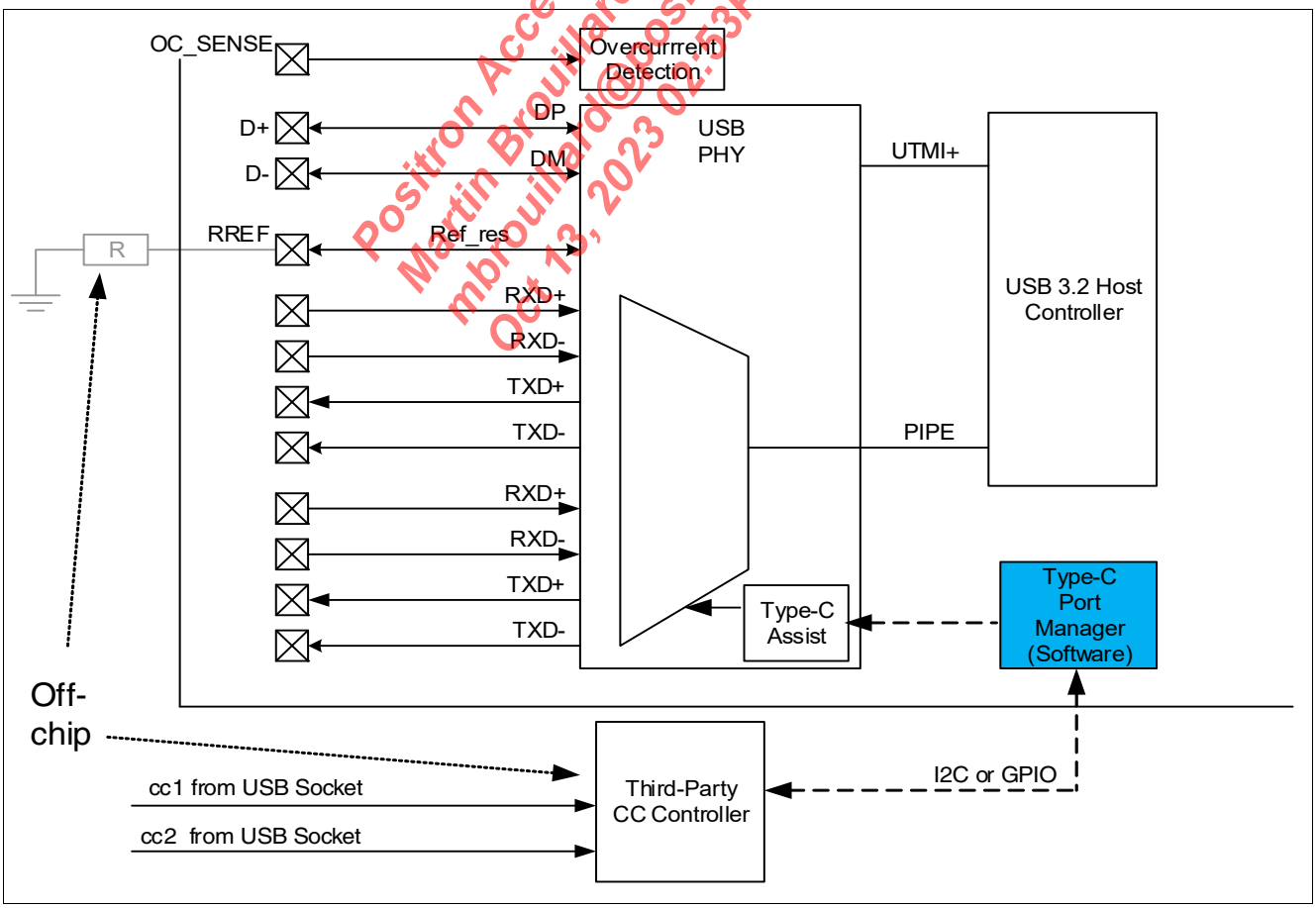


Figure 48 External USB 3.2 Gen 2 Type-C Subsystem Signals

Figure 49 shows a system block design example.

Attention: The MxL25641 only supports lane 0 SERDES functionality and does not support USB Type-C assist functionality.

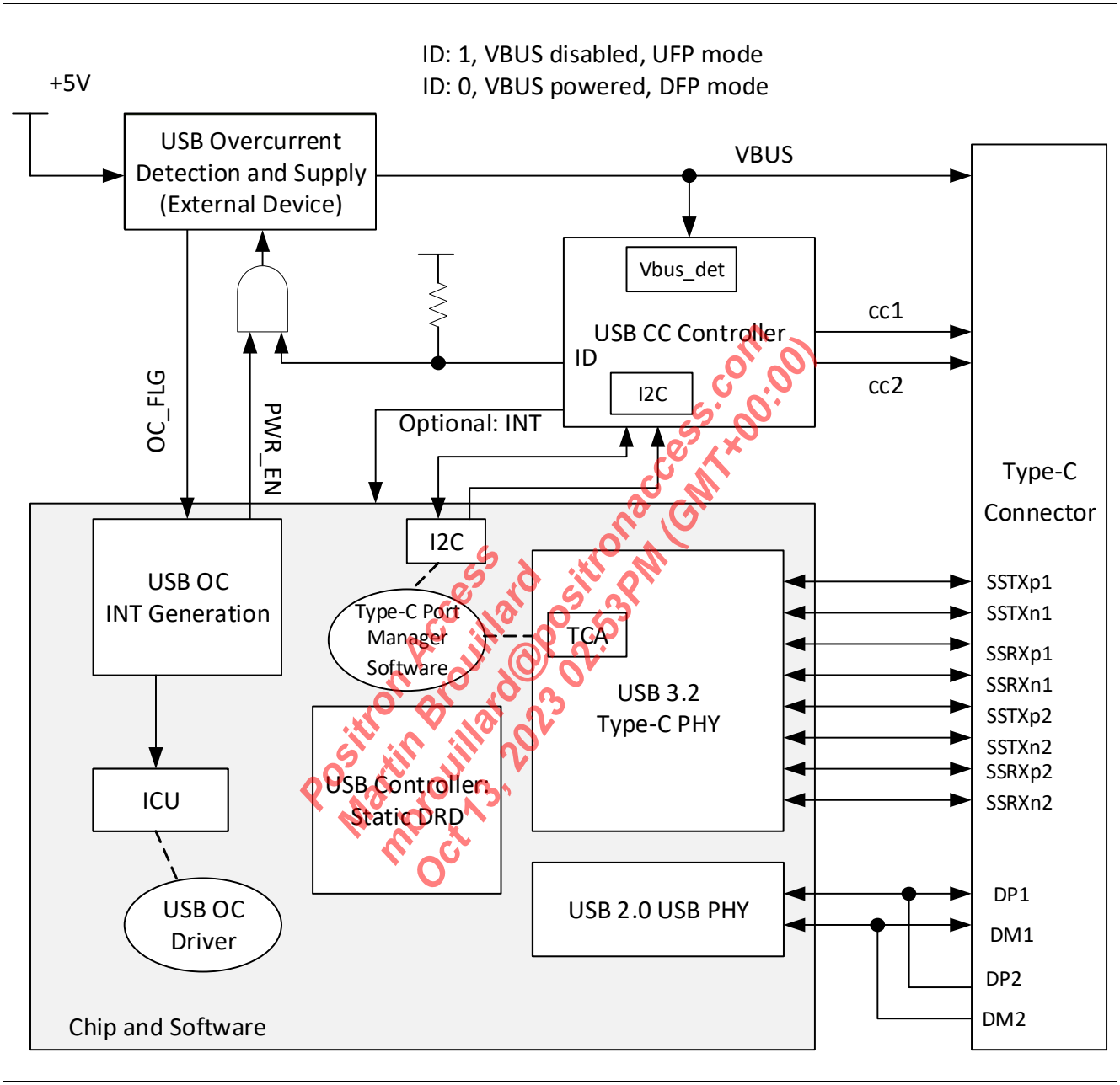


Figure 49 USB Host and Device Mode System Design Example

Attention: The MxL25641 does not support USB Type-C assist functionality. When a USB Type-C is required, it must be implemented with an external IC-based solution.

8 Network Subsystem

The network subsystem description covers these sections:

- **Packet Processing v4 (PPv4)** (**Section 8.1**)
  Handles the network bridging and routing process.
- **Central Queue Manager** (**Section 8.2**)
- **QoS Subsystem** (**Section 8.3**)
- **Buffer Manager** (**Section 8.4**)
- **DMA Controller** (**Section 8.5**)
  Manages packet data movement.
- **Storage Application Acceleration Engine - TCP Offloading Engine (ToE)** (**Section 8.6**)
  Handles TCP, UDP, and MPTCP segmentation and reassembly offloading.
- **Memory Copy DMA Acceleration** (**Section 8.7**)
  Handles application-bonded data copy offloading.

8.1 Packet Processing v4 (PPv4)

The PPv4 subsystem contains these modules:

- Receive Packet Buffer
- Parser
- Classifier
- Traffic Monitor
- Statistics Engine
- Header Modification Engine
- Load Balancer
- For URX851/URX850/MxL25641:
  - 4x egress microcontroller subsystems
  - 4x ingress microcontroller subsystems

8.1.1 Features

The networking subsystem encapsulates the entire SoC networking needs for home and SMB networking, this includes WAN port(s), LAN port(s), and additional ports.

The packet processor is at the heart of the networking subsystem. It accelerates gateway and routing to rates which cannot be met with pure software implementations within a reasonable cost or power.

The PPv4 is a next generation, high performance and low-power Gateway Intellectual Property (GWIP) which combines technology from MaxLinear SLIC for CPE, AnyWAN™ SoC GRX550 Series, and Puma™ 7 Family devices.

The PPv4 subsystem is designed to provide a high bandwidth external WAN connection, which offloads the relevant protocols/stacks from the CPU, and also handle WAN to LAN and LAN to WAN forwarding (based on L2, L3, and/or L4 rules), including offloading Virtual Network Functions (VNF)-based traffic.

This section describes the features of the PPv4.

8.1.1.1 Packet Processing Engine Gen 4 (PPv4)

Table 97 describes the key features of the PPv4 hardware engine.

Table 97 Packet Processing Engine (PPv4)

Features	URX851/URX850	MxL25641
Processing Capability of Core Hardware Engine	30 million packets per second	15 million packets per second
Search Engine Module	Cuckoo hashing-based session entry search	
Packet Memory addressing	32 bits and 36 bits	
Multiple Processing	Supports cycling packets multiple times through the processing flow per session.	
In-order Processing	Maintains the relative order of packets per session to ensure that packets are transmitted in the order in which they were received.	
Maintain Session Packet Order	Synchronizes slow path and fast path packets which correspond to the same session so that no re-ordering is guaranteed up to a pre-defined packet rate.	
Header Analysis	• Ethernet + VLAN (up to two stacked VLANs) • PPP over Ethernet (PPPoE) protocol • Transmission Control Protocol (TCP) • User Datagram Protocol (UDP) • IPsec headers • IGMPv2/IGMPv3 (up to 512 multi-cast groups) • ICMP • L2TPv2/L2TPv3 tunneling • 6rd tunneling support • DS-Lite tunneling • VXLAN tunneling • GRE and NVGRE tunneling • Stateless Transport Tunneling (STT) • MAP-t and MAP-E tunneling • Up to three stacked tunnel protocols • IPv4 and IPv6 and IP in IP • Additional headers are supported by parser configuration and/or microcontrollers.	
Modification	• The modification logic supports these basic operations: – Insert a string *s* at location *i* in a packet. – Delete a string of length *l* from location *i* in a packet. – Replace a string of length *l* at location *i* with *s* in a packet. – Perform L3 and L4 checksum delta recalculation at location *i*. • Supports chaining basic operations. • Supports performing at least 6 basic modifications per packet where the values of *i*, *n*, *l*, and *s* are defined per session per modification operation. • Locations relative to specific headers (like header checksum) are given relative to the header pointer as extracted during the frame parsing.	
Session	• Supports any number of sessions up to 16M (The system configures 64K as default). • Session string is configurable to up to 128 bytes. • Associates a session to modification rules, counters, and QoS queue. • Associates a session with a drop policy to enable firewall rules acceleration. • No dependency of performance in locality in time of sessions • Supports hardware-assisted pruning of sessions to lower CPU load.	

Table 97 Packet Processing Engine (PPv4) (cont'd)

Features	URX851/URX850	MxL25641
Exception Handling	• When a packet does not match any configured sessions, the packet process classifies the packet based on the incoming port, Traffic Class (TC). A 2-bit hash is performed on the Field Vector (FV) and assigned to an exception session. It is possible to define up to 63 exception sessions. • The support session exception diverts packets from the regular classifier-modifier-QoS port flow to an alternate flow. This alternative flow can be a stateless microcontroller, a stateful microcontroller, the Intel Atom CPU subsystem, or the packet is discarded. These exceptions are supported: – Packets with size larger than the configured egress port MTU – TCP control packets – Packets with TTL/HOP limit expired – Packets with IPv4 options or IPv6 unsupported extension headers – Fragmented packets – Multicast packets – TCP ACK packets	
Flow Control and Rate Monitoring Features of the PPv4		
Flow Control and Rate Monitoring	• Early drop at ingress based on port packet coloring – Ingress QoS based on TC provided by pre-parsing – Ingress QoS can perform packet drop or use a flow control mechanism. • Traffic monitors may be applied to re-color packets based on the rate of the traffic, so congestion is handled in a mindful manner. The color is eventually used to aid in congestion avoidance. Packet drop policies are color-aware. • Supports traffic rate monitoring. • Supports 256 dual rate monitors. Each session may be associated with up to five cascaded-monitors of arbitrary order. • The traffic rate monitor may be referenced by any number of sessions. • Provides a policy per traffic rate monitor such that when packets associated with the monitor exceed a pre-configured rate, re-coloring of the traffic occurs.	
Statistics and Debug Features		
RMON Counters	• 40-bit packet counters per session • 54-bit byte counters per session • Packet and byte counters per color at egress (after final coloring decision) • Packet and byte group counters that can be associated with group of sessions. Group counters are divided into 8 different pools. The pool sizes are: – 2x 1024 counters – 1x 512 counters – 1x 256 counters – 2x 64 counters – 1x 16 counters – 1x 8 counters There are a total of 2,952 counters. • Supports associating up to eight group counters (one per pool) to a session. The association of sessions to groups is done by the host software per session. • Supports associating any number of sessions with a group counter.	

8.1.1.2 Flexible Processing Engine (FPE, part of PPv4)

Table 98 describes the key features of the FPE.

Table 98 Acceleration Modules - Flexible Processing Engine (FPE)

Features	URX851/URX850	MxL25641
Ingress FPE Functions	Parses complex protocols unsupported by the hardware parser: • Pre-modification for irregular protocols unsupported by the modifier • Pre-post classification in addition to the existing hardware engine	
Egress FPE Functions	Performs multi-cast, fragmentation, reassembly, ACK suppression functions (512 sessions).	
Session Search	Hardware-based hashing and search	
Routing Entries	Up to 16 million entries	
Embedded Processor	• Four Ingress • Four Egress • ARC 32-bit RISC with DMA accelerator	

8.2 Central Queue Manager

The CQM is composed of various submodules:

- The Enqueue Manager (EQM) is responsible for the assignment of free buffers to the ingress ports and en-queuing the filled buffers into the QoS engine queues.
- The Dequeue Manager (DQM) is responsible for dequeuing the buffers from the QoS engine queue to the egress ports and return the free buffers to the Buffer Manager (BM).
- The Free Segment Queue Manager (FSQM) is responsible for managing the buffers in internal memory and for linking segments in segmented internal packets.
- The WAN Interface Block (WIB) is responsible for taking in requests from the PON-IP or the DOCSIS sidecar for dequeuing the required number of packets out of dequeue manager for the respective upstream queues.
- The Load Spreader (LS) is responsible for spreading the egress traffic towards the CPU.
- The UMT counter block counts the Tx and Rx counters for the AnyWan™ Common Architecture (ACA) ports and periodically sends these counters over the PCIe bus to the ACA devices.

8.2.1 CQM Specification

The CQM is defined in this section.

The CQM provides buffers from 2 locations: the internal packet buffer, which is 512 KB and from external DDR. This example configuration demonstrates an allocated buffer pool:

- 8 k buffer pointers for internal packet buffer (128B) segments. Option to have 512 buffers of size 2 kB each
- 256 B (size0) buffer pointers for external buffers
- 1 kB (size1) buffer pointers for external buffers
- 2 kB (size2) buffer pointers for external buffers
- 10 kB (size3) buffer pointers for external buffers
- 4 kB (size4) buffer pointers for the 8 VM pools. One pool per VM of the same size
- Direct Connect-capable peripherals can use a buffer from any of the four fast path pools depending on their configuration in the BM. This is programmable. The software must configure the corresponding size for the Direct Connect pool.
- The number of external buffers is programmable and set up during the initialization of the system. The BM block is also able to handle this.

The FSQM handles the internal buffer pool where as the BM manages the external buffer pool. A buffer pointer is used per packet or can be part of a segment of a packet and the decision is based on the start of packet and end of packet bits in the packet descriptor. Multiple segments per packet are supported for internal buffers. Each set of DMA descriptors consists of a FIFO of 8 descriptors to minimize latency in the enqueue and dequeue process of the QoS Engine.

8.2.1.1 Enqueue Manager

The EQM must be generic to support up to 64 ports for future applications. Each port must be made generic to function as a CPU port or DMA port. The URX85x supports up to 34 ports. These ports may be allocated according to these criteria:

- Up to 16 CPU ports may be used for these CPU-based clients:
  - Four ports for the networking processor
  - Eight ports for the Virtual Machine (VM) functions
  - Four ports for the voice CPU
  - Each CPU port has five buffer request registers (4 for external buffer pools and 1 for the internal buffers) and 1 descriptor enqueue register set for enqueue of both segmented internal or single internal/external packets.
- Up to 10 DMA ports may be used for these DMA-based clients:
  - Seven ports for DMA0-Rx on one FSM
  - Two ports for TOE Engine (one each for TSO and LRO)
  - One special port for the PPv4 Rx DMA to request and enqueue from the FSQM managed buffers
- Up to eight Direct Connect ports, with separate descriptors for segmented and non-segmented traffic, supporting ACA peripheral support
- The VPN adapter uses Direct Connect ports to support enqueue requests from the flexible descriptor converter module

Table 99 summarizes the port configuration for URX85x devices.

Table 99 Ingress Port Configuration URX851/URX850

Ingress Module		CPU	DC	DMA1 Rx	DMA-TOE Rx	PPv4 Rx-DMA
CPU Ingress Ports	Standard	16	–	–	–	–
	Special (DC)	–	8	–	–	–
DMA Ingress Ports	Standard	–	–	7	2	–
	Special	–	–	–	–	1

Each normal DMA port supports a list of descriptors for internal or external buffer pointers based on the policy of the port. Each list of descriptors supports FIFO size of 8 descriptors of 4 DWORDs each.

Each standard CPU port supports:

- FIFO size of 2 descriptors of 4 DWORDs each

Each DC port supports:

- FIFO size of 32 descriptors of 4 DWORDs each

There is only a common descriptor list to enqueue for both internal and external packets. In addition to these descriptors, each CPU port also has 5 buffer request registers, 1 for the internal buffer packets and 4 for the external buffer packets. The DC ports have a ring of 16 free buffers for 1 buffer size which is configured via the DC port configuration. Each buffer request register is 36 bits. **Section 8.2.1.5** shows the format.

Table 100 summarizes the size of the descriptor list and buffer request.

Table 100 Ingress Port Descriptor and Request Configuration

Ingress Port	CPU	DC	DMA	
	Standard	Standard	Standard	Special
Standard Enqueue Descriptors	2[1] (4 DW)	32[1] (4 DW)	8 (4 DW)	16 (2 DW)
Standard Buffer Request	5x4+1x12 (per CPU port type)	16x8	–	8
Repeat Count Request	–	–	–	–

1) For either internal or external packet

The enqueue manager works in a round-robin matter from one port to the next with the same priority, regardless of CPU port or DMA port. It serves one port at a time and it skips ports without enqueue requests.

All DMA and CPU ports share a common interface to the FSQM. All CPU ports buffer request have independent interface to FSQM.

Table 101 summarizes the ingress port mapping to the ingress modules.

Table 101 Ingress Port Mapping

Port Type	Port ID	Ingress Module URX851/850	MxL25641
CPU	0-3	Networking CPU 0-3	Networking CPU 0-1
	4-11	VM 0-7	VM 0-3
	12-15	Voice 0-3 (Use Policy4 reg)	Voice 0-1 (Use Policy4 reg)
DC	16-23	DC 0-7	DC 0-3
DMA[1]	24	DMA0Rx Ch0 (Internal, Reserved)	DMA0Rx Ch0 (Internal, Reserved)
	25	DMA0Rx Ch1 (External, Size0)	DMA0Rx Ch1 (External, Size0)
	26	DMA0Rx Ch2 (External, Size1)	DMA0Rx Ch2 (External, Size1)
	27	DMA0Rx Ch3 (External, Size2)	DMA0Rx Ch3 (External, Size2)
	28	DMA0Rx Ch4 (External, Size3)	DMA0Rx Ch4 (External, Size3)
	29	DMA0Rx Ch5 (Header Mode)	DMA0Rx Ch5 (Header Mode)
	30	DMA0Rx Ch6 (External, Size2)	DMA0Rx Ch6 (External, Size2)
	31	TOE-TSO (Use Policy4 reg)	TOE-TSO (Use Policy4 reg)
	32	TOE-LRO (Use Policy4 reg)	TOE-LRO (Use Policy4 reg)
	33	PPv4 Rx-DMA Special port	PPv4 Rx-DMA Special port

1) The port mapping is applicable only for DMA in on-demand mode.

8.2.1.2 Dequeue Manager

The DQM must be generic and support up to 256 ports for future applications. Each port must be made generic to function as a CPU, DMA, or PON port. The URX85x supports up to 139 ports. It is possible to allocate these ports according to these criteria:

- Up to 19 CPU ports
  - Eight ports for the networking core. One high priority and one low priority for spreading.
  - Eight Ports for the VM with one port per VM
  - Two special ports for the voice subsystem
  - One port for the LRO
- Up to 48 DMA ports supporting normal mode for these DMA clients
  - 16 ports for DMA0-Tx
  - 16 ports for DMA1-Tx
  - 16 ports for DMA2-Tx
- Up to eight Direct Connect ports supporting ACA peripheral support
- Up to 64 special PON ports, with separate descriptors for segmented and non-segmented traffic, for the PON-IP IF block
  - 64 ports for PON-IP via the PON-IP IF block
  - The DMA connects to these ports on a single channel via the PON-IP IF

Table 102 to Table 104 summarize the port configuration.

Table 102 Egress Port Configuration URX851

Egress Module		CPU	DC	DMA2 Tx	DMA1 Tx	PON IP	DMA0 Tx
CPU Egress Ports	Standard	17	8	–	–	–	–
	Special	2	–	–	–	–	–
DMA Egress Ports	Standard	–	–	12	12	–	8
	Special	–	–	4	4	–	8
PON Egress Ports	Standard	–	–	–	–	64	–
	Special	–	–	–	–	1	–

Table 103 Egress Port Configuration URX850

Egress Module		CPU	DC	DMA2 Tx	DMA1 Tx	PON IP	DMA0 Tx
CPU Egress Ports	Standard	17	8	–	–	–	–
	Special	2	–	–	–	–	–
DMA Egress Ports	Standard	–	–	12	12	–	8
	Special	–	–	4	4	–	8

Table 104 Egress Port Configuration MxL25641

Egress Module		CPU	DC	DMA2 Tx	DMA1 Tx	PON IP	DMA0 Tx
CPU Egress Ports	Standard	9	4	–	–	–	–
	Special	1	–	–	–	–	–
DMA Egress Ports	Standard	–	–	12	12	–	8
	Special	–	–	4	4	–	8
PON Egress Ports	Standard	–	–	–	–	64	–
	Special	–	–	–	–	1	–

Each normal DMA port supports a list of descriptors. The list of descriptors supports a ring size of 8 descriptors consisting of four DWORDs each.
The format of these descriptors is the same as the format defined for the central DMA Tx descriptors.
Each standard CPU port supports a FIFO size of two descriptors consisting of four DWORDs each.
There is a common descriptor list for all packets, regardless of size. In addition to these descriptors, each CPU port has a buffer return register for all packets, regardless of size. Each buffer return register is two DWORDs.
Each special CPU port for the voice subsystem supports a FIFO size of four descriptors consisting of four DWORDs each.
Each PON-IP IF port supports a list of descriptors for the packet descriptors. The list of descriptors supports FIFO size of eight descriptors consisting of four DWORDs each. This list contains only a packet descriptor and not segment descriptors. There is a separate set of descriptor list containing the segmented descriptor list of the first descriptor in the packet descriptor list which identifies a segmented packet with an end of packet value of 0. This list has a depth of eight descriptors.
There is a single buffer return register for all PON-IP ports. This register behaves the same as the CPU buffer return register.
Table 105 summarizes the size of the descriptor list and buffer request.

Table 105 Egress Port Descriptor and Request Configuration

Egress Port	CPU		PON IF		DMA	
	Standard	Special (DC)	Standard	Special	Standard	Special
Dequeue Descriptors	2 (4 DW)	32 (4 DW)	8 (4 DW)	–	8 (4 DW)	8 (4 DW)
Buffer Return Request	1	32	–	1	–	–
Dequeue Segmented Descriptors	–	–	8	–	–	–

The dequeue manager works in a round-robin manner from one port to the next with the same priority. It serves one port at a time and it skips ports without dequeue requests. The CPU, DMA, and PON-IP ports work in parallel.
Table 106 summarizes the egress port mapping to the egress modules.
The port ID represents the CQM DQM port ID and not the QoS port ID. The QoS port to DQM port ID mapping is performed via the configuration of the EPMAP in the CQM and the correct ring address is configured in the Tx-Manager. The DMA0Tx Ch0 is mapped to both the DMA port the PIB port. The descriptor ready for both these ports must be ORed/ANDed and provided.

Table 106 Egress Port Mapping

Port Type	Port ID	Egress Module URX851	Egress Module URX850	Egress Module MxL25641
CPU (Standard)	0-7	Networking CPU0-7	Networking CPU0-7	Networking CPU0-3
	8-15	VM 0-7	VM 0-7	VM 0-3
	16	LRO	LRO	LRO
	17-18	Voice 0-1 (Special)	Voice 0-1 (Special)	Voice 0 (Special)
DC	19-26	DC 0-7	DC 0-7	DC 0-3
DMA[1)] (Standard)	27-42	DMA2Tx Ch0-15	DMA2Tx Ch0-15	DMA2Tx Ch0-15
	43-58	DMA1Tx Ch0-15	DMA1Tx Ch0-15	DMA1Tx Ch0-15
	59-74	DMA0Tx Ch0-15	DMA0Tx Ch0-15	DMA0Tx Ch0-15
PON-IP	75-138	PON IP IF (DMA0Tx Ch0)	-	PON IP IF (DMA0Tx Ch0)

1) The port mapping is applicable only for DMA in on-demand mode.

It is possible to enable and disable each port individually. All internal states related to each port start from a clean reset state upon enabling the port. However, all registers and counters related to the port are not reset upon enabling the port. The software must reprogram them to desired values before enabling the port.

8.2.1.3 Buffer Manager and QoS Engine Commands

The EQM and the DQM interact with the BM and the QoS engine blocks from the PPv4 engine. These interactions happen over the PPv4 SSX as commands to and from these blocks.

The EQM interacts with the BM for:

- Allocating free buffers from the external buffer pools managed by the BM.
- Returning back free buffers when an enqueued packet is discarded.
- Returning back free buffers when a CPU port allocation does not meet a matching PortID and PolicyID criteria.

The EQM interacts with the QoS engine for:

- Issuing a WRED query command to the QoS engine WRED block.
- Receiving the response from the WRED query command.
- Issuing a Q-Push command to push the descriptor into the Q-Manager upon a successful WRED acceptance.

The DQM interacts with the BM for:

- Returning back free buffers after a DMA Tx transaction completes.
- Returning back free buffers when a buffer return command is written by CPU or ACA ports.

The DQM interacts with the QoS engine for:

- Pushing a descriptor to be dequeued from the Tx Manager in the QoS engine.
- Replenishing credits back to the Tx manager once a descriptor is successfully dequeued from the DQM.

Note: The QoS engine only supports a single descriptor per packet and thus does not support segmented buffers. It is the responsibility of the EQM to combine segmented descriptors into a single descriptor which has the complete packet size. The EQM must also keep the end of packet bit in the descriptor as 0 to identify the descriptors as segmented packet and indicate the DQM that it needs to recreate the list of segmented descriptors from the linkage created by the EQM in the LLT memory of the FSQM. Thus, for the QoS Engine, each packet is represented by a single descriptor of four DWORDs (128 bits).

8.2.1.4 WAN Interface Block

The dequeuing function of the PON WAN and DOCSIS interface requires special logic to manage the prefetch of expected queues from the QOS block. For PON, it is referred by the GEM ports to be transmitted into the playout buffer of the PON IP. This requires a special block in between the dequeue manager and the PON IP to translate the commands from the PON IP to the relevant dequeue commands to the CQM and then to present the dequeued descriptors to the GSWIP-O DMA to fill the playout buffer.

Figure 50 shows the top level view of the interaction between the PON IP and the CQM DQ-M. The PON IP reads the backlog counters maintained per egress port of the QoS engine (dequeue ports) and per ingress queue of the QoS Engine to understand the amount of data which is queued up in the various queues for each TCONT. Each TCONT is represented by a port on the DQ-M. Based on the values of the backlog counters and the previous transmissions, the PON IP requests a specific amount of data from each TCONT. The WIB receives these requests and process them in order. It dequeues the packets from the DQ-M port and provides them on the DMA port for the GSWIP-O to read.

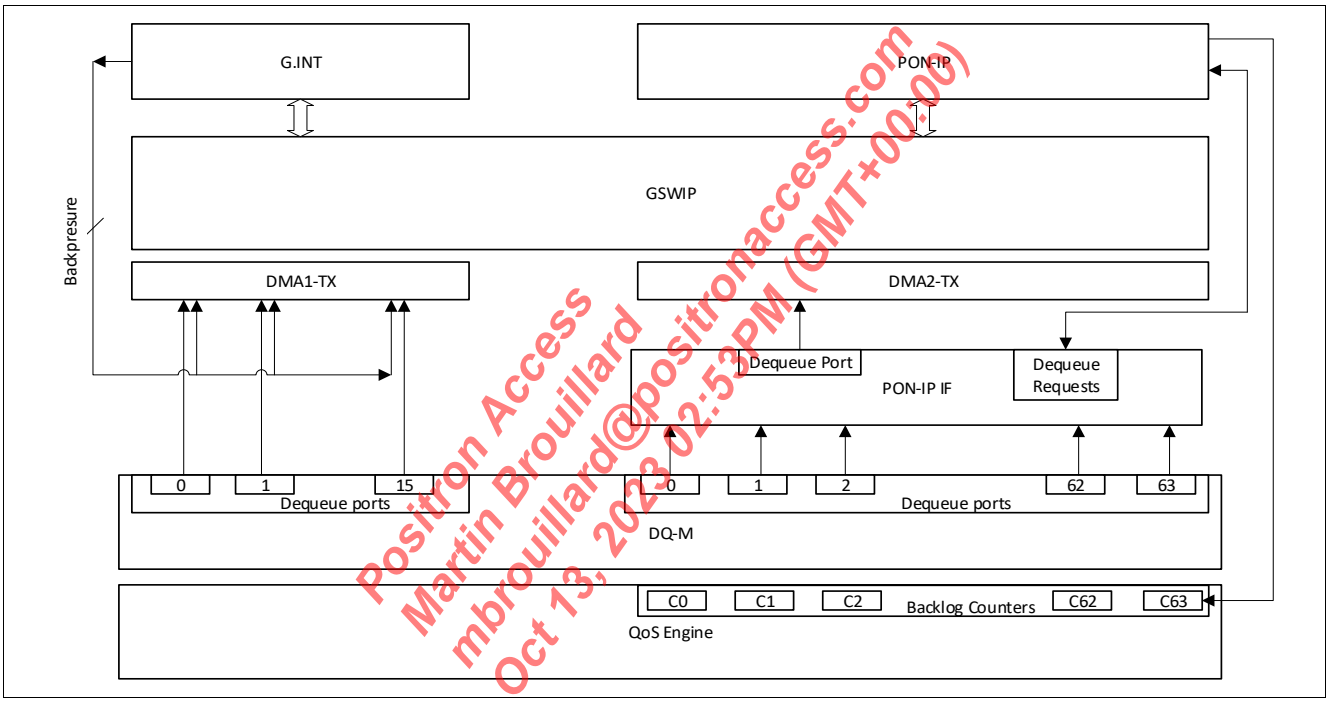


Figure 50 WAN Interface Block

8.2.1.5 Descriptor Formats

The CQM blocks handles three types of descriptor formats.

Figure 51 shows the QoS descriptor format, the command to return back the free buffer by the CPU port, and the PIB port.

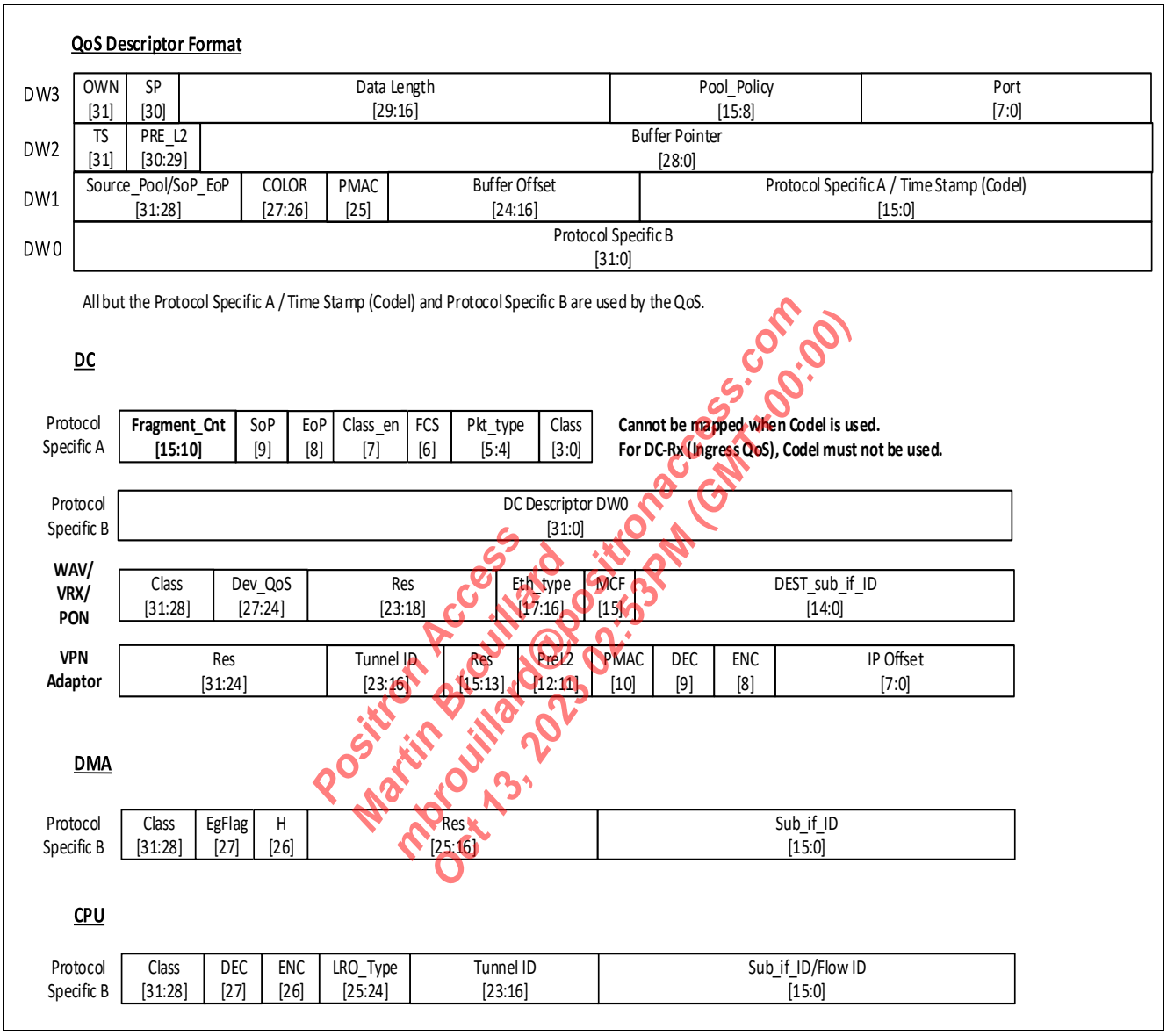


Figure 51 QoS Descriptor and Buffer Return Command

Figure 52 shows the DMA descriptor format. The CQM must provide the DMA descriptor in the format described. The CQM must extract the required fields to compute the QID.

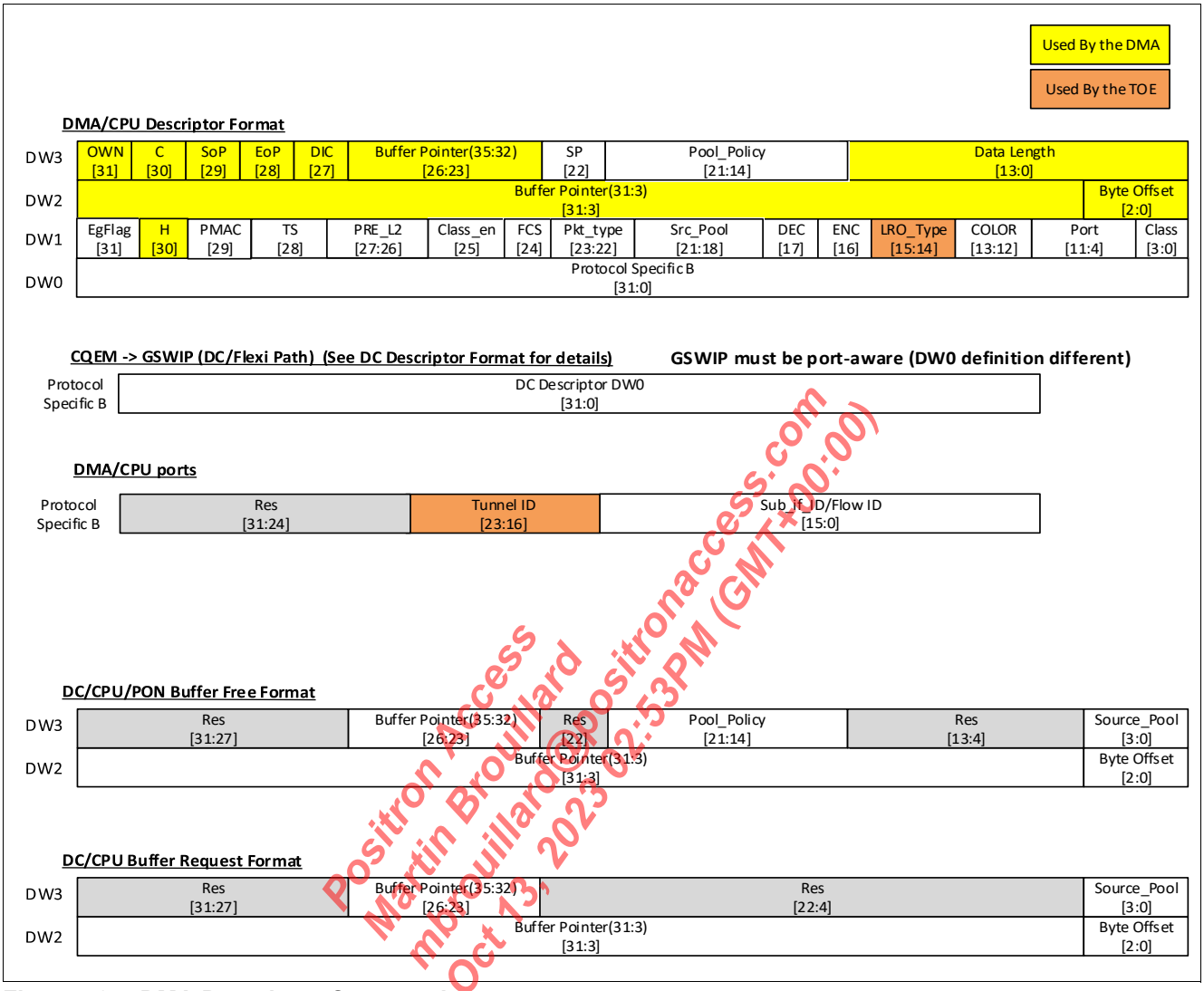


Figure 52 DMA Descriptor Command

The PPv4 RxDMA interacts with the CQM for the exchange of free buffers and to link segments when it needs to use the SSB for segmented memory. **Figure 53** shows the enqueue descriptor format. The CQM must provide the DMA descriptor in the format described. The CQM must extract the required fields to link segments in the FSQM. The port behaves like a CPU port with different buffer request and enqueue registers.

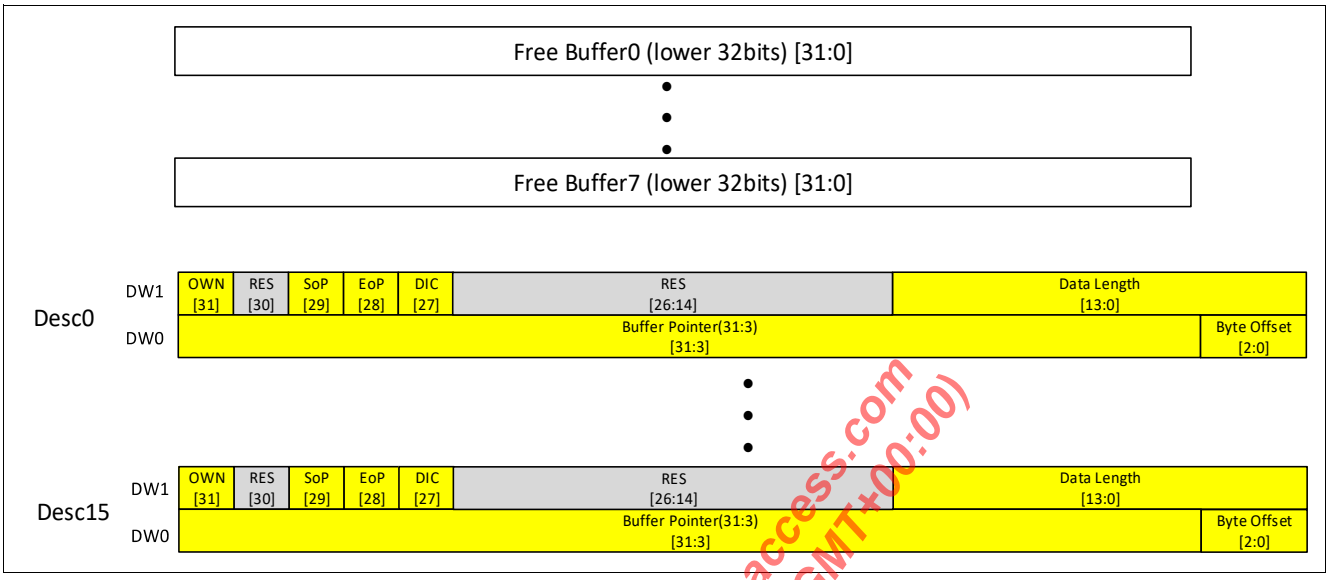


Figure 53 PPv4 Rx DMA Descriptor Command

Figure 54 shows the Direct Connect descriptor format. The CQM must provide the DC port descriptor in the format described. The CQM must extract the required fields to compute the QID. The CQM must also convert between DC descriptor and QoS descriptor formats when performing enqueuing and dequeuing operations.

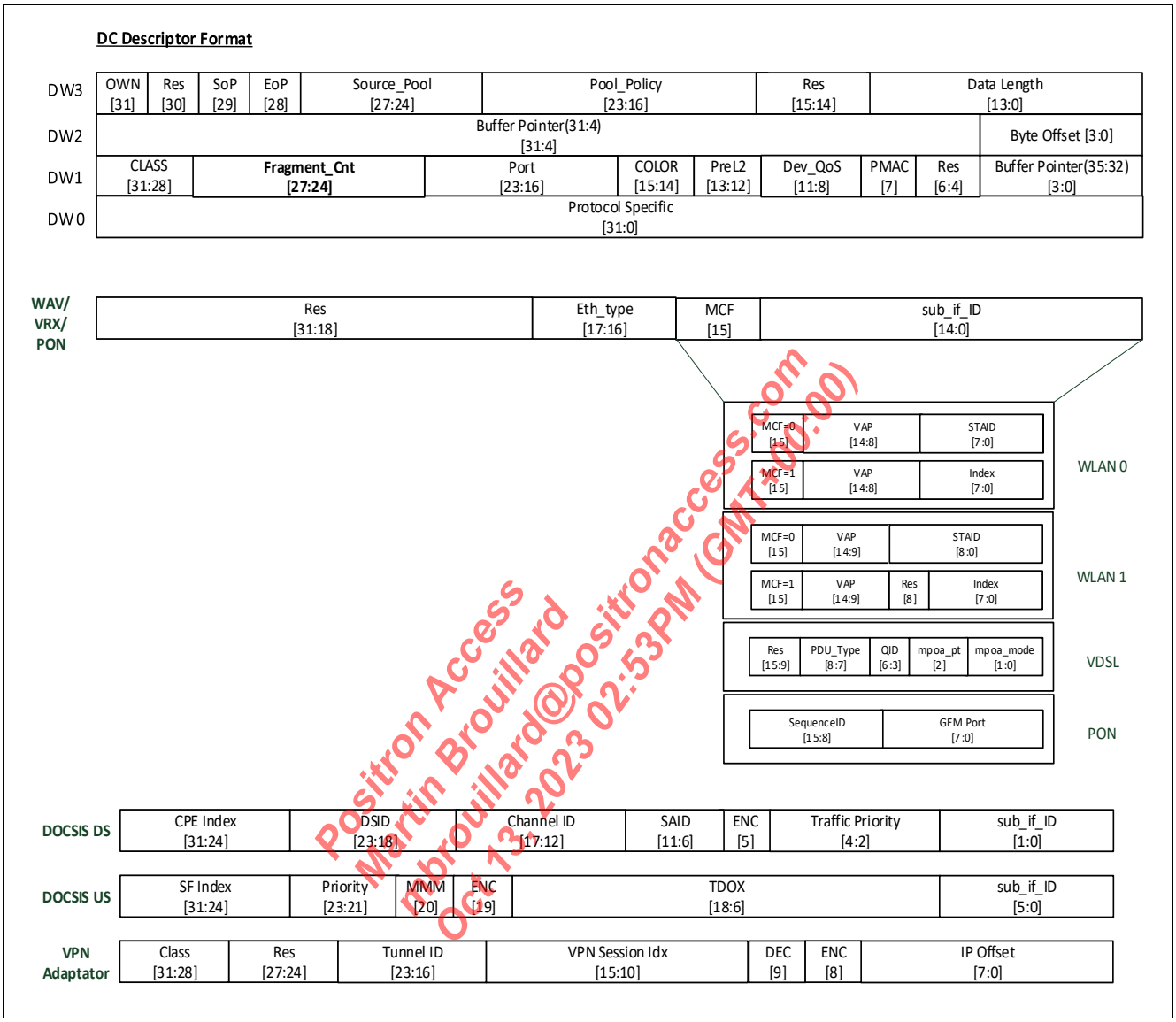


Figure 54 DC Descriptor Command

8.2.2 PON-IP

The URX851, and MxL25641 devices are a Transmission Convergence (TC) layer or Physical Coding Sublayer (PCS) implementation which integrates SoCs between a link layer processing instance and a physical layer interface. **Figure 55** illustrates this integration.

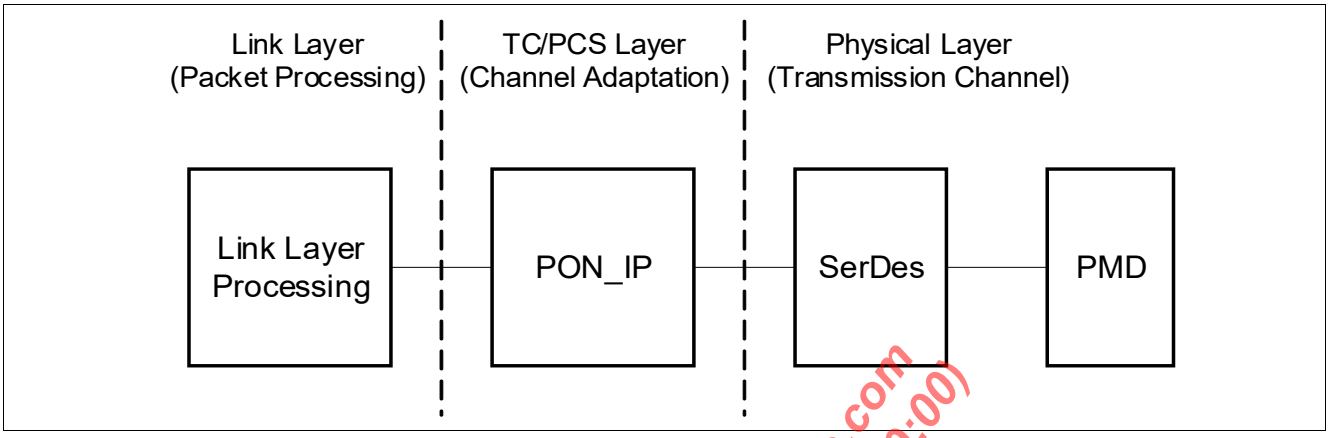


Figure 55 URX851/MxL25641 Integration Overview

The PON subsystem consists of these functions:

- Physical layer
- Link layer including flow control
- SoC infrastructure such as clocking, reset, configuration access, and similar components
- Management

This section discusses different aspects of link layer integration.

8.2.2.1 Sublayer Structure

Figure 56 shows the boundaries within the URX851/MxL24641 between the Link Layer in ITU (GPON) and IEEE (EPON) modes.

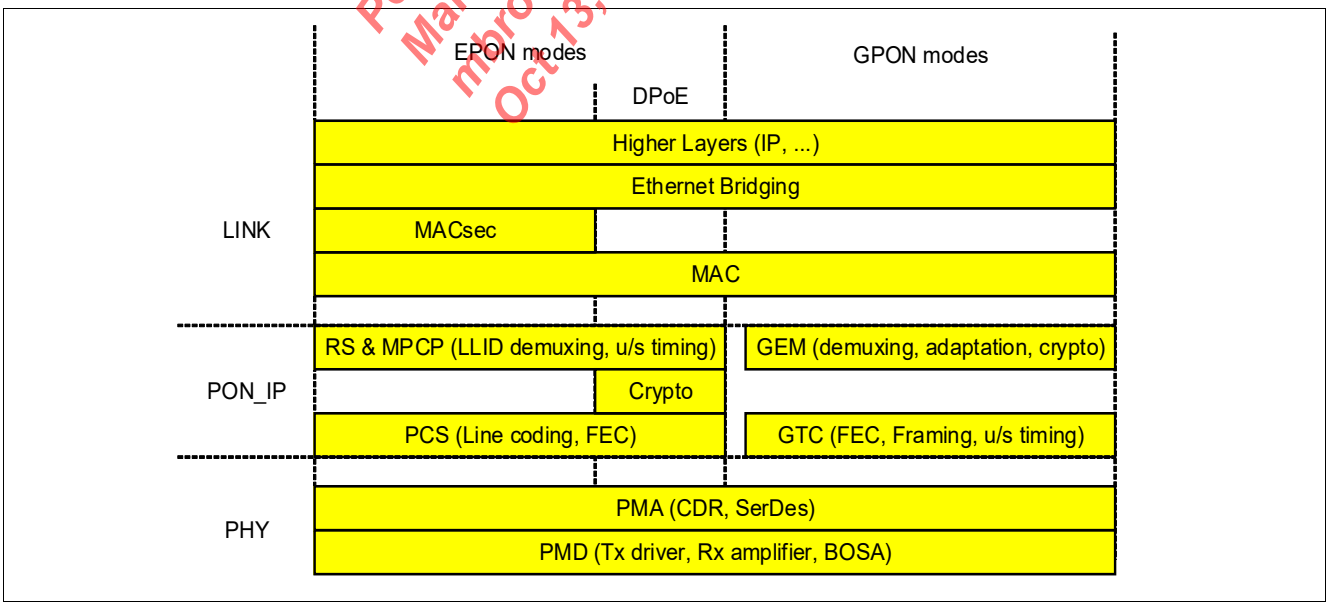


Figure 56 Sublayer Structure

8.2.2.2 PON WAN Throughput

The net data rates at the interface between URX851/MxL25641 devices and the link layer are configuration dependent. These factors influence the net data rate:

- Standard: GPON, 10GPON, EPON, Turbo EPON, or 10G EPON
- Connection type: symmetric or asymmetric
- Forward Error Correction (FEC): enabled or disabled
- The amount of overhead from GTC, PLOAM, or MPCP

8.2.2.2.1 10G GPON

Baud rate = bit rate = 9.95328 Gbit/s

GEM layer bit rate = 8.66 Gbit/s when FEC is enabled; 9.95 Gbit/s when FEC is disabled.

IPG = 8 bytes (XGEM header)

8.2.2.2.2 10G EPON

Baud rate = 10.3125 Gbaud/s

Bit rate = 10.0 Gbit/s

IPG = 20 bytes

8.3 QoS Subsystem

The QoS subsystem is responsible for allocating of egress bandwidth (of physical as well as virtual ports) according to a configurable and controllable scheme, allowing prioritization of traffic, fairness, and rate limiting.

The QoS subsystem follows the packet processing phases. It consists of ingress queues where packets are inserted by the packet processing stage and egress queues representing the physical or virtual ports.

8.4 Buffer Manager

The buffer management subsystem manages the network buffers. It provides allocation and recycle services to networking IPs. Buffer pools are data structures holding a collection of buffers of the same size and used for storing network packets.

The purpose of the buffer manager is to allocate buffers from buffer pools and recycle them when no longer needed. To be both efficient in memory utilization and fast, the buffer manager performs accounting of allocations per ports and pools. This accounting enables the buffer manager to apply policies that guarantee that buffers are utilized in an efficient, yet fair way. The main goals of the buffer manager are:

- Provide efficient buffer allocation and recycling
- Maximize internal RAM use while minimizing DDR RAM use.
- Utilize DDR memory efficiently
- Manage resources in accordance with fairness and resource reservation policies to guarantee a minimum are available

8.5 DMA Controller

The DMA controller description covers these sections:

- **V2.x Features** (**Section 8.5.1**)
- **V3.1 New Features** (**Section 8.5.2**)
- **DMA Instances and Configurations** (**Section 8.5.3**)

8.5.1 V2.x Features

The DMA controller provides these features:

- A DMA controller in general
  - An Rx controller and a Tx controller which operate independently
  - One Tx FIFO shared between all Tx channels
  - One Rx FIFO shared between all Rx channels
  - Arbitrates memory controller access between Tx, Rx, and the descriptor manager
- DMA controller to memory
  - 32/64-bit wide data path
  - 332-bit configurable data pointer pointing to 30-bit word addresses which results in 2^{32} = 4096 Mbytes address range (descriptor data pointer)
  - 16-bit data length allowing for a maximum of 2^{16}= 64 k transferable bytes (descriptor data length)
  - Programmable byte offset of 0 up to 7 bytes at the start of packet for Tx (64-bit DMA), 0 up to 3 bytes at the start of packet for Tx (32-bit DMA)
  - Programmable byte offset of 0 up to 7 bytes at start of packet for Rx (64-bit DMA), 0 up to 3 bytes at the start of packet for Rx (32-bit DMA)
  - Programmable transfer sizes of 4 x 32/64-bit or 8 x 32/64-bit, 16 x 32/64-bit
- DMA controller to peripherals
  - DPlus port to interconnect peripherals to the DMA controller
  - Possible memory to memory (memcpy) port, no peripheral connection
  - Programmable burst transfer sizes of 4 x 32/64-bit or 8 x 32/64-bit, 16 x 32/64-bit
- Endianness
  - Programmable byte swapping functionality
  - Individually programmable in the Tx and Rx direction for each peripheral port
- Arbitration scheme
  - Arbitration individually performed for DMA channels in the Tx and Rx direction
  - Rx arbitration is based on a round robin arrangement.
  - Tx arbitration is based on weight values among ports and channels.
  - Arbitration on burst boundaries between different peripherals (ports) of the same priority
  - Arbitration on packet boundaries between DMA channels within the same peripheral (port)
- DMA descriptors
  - Rx and Tx descriptor lists are stored in memory.
  - One descriptor entry per DMA channel is stored within the DMA (Descriptor Manager).
  - Each descriptor list controls the operation of one DMA channel.
  - A descriptor list consists of at least one descriptor entry.
  - The number of descriptor entries per descriptor list is specified individually for each DMA channel, with a maximum of 4,095 entries..
  - A descriptor list entry must be located at a 64/128-bit aligned address.
  - Descriptors are concatenated as a linear structure in continuous memory locations.
  - The width of a descriptor entry is 64/128-bit and it contains data length, data pointer, byte offset bit fields, status, and control bits.
- Interrupt Generation
  - Each DMA channel has five interrupt request sources.

  - Each DMA channel has its own interrupt mask and interrupt status register.
  - Each DMA instance also have one combined interrupt which ORs all channel interrupt.
- FPI Bus Slave for processor control and set-up
- Supports single packets multiple descriptors
- On demand channel activation, this is possible before the DMA starts to transfer data.
- Activity based power saving feature
- Scatter gather DMA support
- Packet drop feature
  - Support of descriptor pre-load when the current Tx data path is active for a single DMA port configuration. Preload can happen to same channel or different channel. This feature works with the DMA descriptor and has a separate OCP port.
  - `Channel_eop_irq_o` indicates whenever the DMA controller has completed processing a packet.
  - Dynamic burst read
  - Configurable descriptor size (2 DWORDs or 4 DWORDs)
  - Descriptor metadata copy and descriptor reserved bit copy for memcpy port

8.5.2 V3.1 New Features

The DMA controller version 3.x provides these advanced features:

- Header only transfer mode
  - Tx header only
  - Rx header only
  - Tx full packet data in Tx header mode
- Scatter gather DMA support for byte offset of every fragment; and every fragment size can be not multiple of burst size.
- OPC master port generates Mreqinfor signals for DMA channel indication
- Descriptor Fetch timeout mechanism
- Descriptor write can be posted write
- DMA Rx supports dynamic burst size instead of static burst size of 2, 4, 8, or 16 DWORDs
- DMA Rx WRNP of last beat to resolve potential descriptor and data update out of sync system issue

8.5.3 DMA Instances and Configurations

Table 107 in this section lists the number of DMA logic channels for peripherals.

Table 107 DMA Logical Channels for Peripherals

Peripheral	Number of Rx Channels	Number of Tx Channels
DMA0-Tx (128 bits data and 36 bits addr,V3.1) Logical Channels for Peripherals		
GSWIP-O PMAC0	0	16
DMA1-Rx (128 bits data and 36 bits addr,V3.1) Logical Channels for Peripherals		
GSWIP-O PMAC1	8	0
DMA1-Tx (128 bits data and 36 bits addr,V3.1) Logical Channels for Peripherals		
GSWIP-O PMAC1	0	16
DMA2-Tx (128 bits data and 36 bits addr,V3.1) Logical Channels for Peripherals		
GSWIP-O PMAC2	0	16
DMA3-MCPY (64 bits data and 36 bits addr,V3.1) Logical Channels		
DMA3-MCPY	8	8 (memcpy)
TOE0-DMA (64 bits data and 36 bits addr ,V3.1) Logical Channels		
TOE-MCPY 0	16	16
TOE-MCPY 1	16	16
TOE1-DMA (64 bits data and 36 bits addr ,V3.1) Logical Channels		
TOE-MCPY 0	16	16
TOE-MCPY 1	16	16
Peripheral DMA0 (32 bits data and 32 bit addr,v2.2) Logical Channels for Peripherals		
SPI0	1 (ch0)	1 (ch1)
SPI1	1 (ch2)	1 (ch3)
SPI2	1 (ch4)	1 (ch5)
SPI3	1 (ch6)	1 (ch7)
HSNAND	1 (ch8)	1 (ch9)
MCPY for PCM0	2 (ch10,12)	2 (memcpy,ch11,ch13)
MCPY for PCM1	2 (ch14,16)	2 (memcpy,ch15,ch17)
MCPY for PCM2	2 (ch18,20)	2 (memcpy,ch19,ch21)

8.6 Storage Application Acceleration Engine - TCP Offloading Engine (ToE)

This section describes the functionality of the storage application acceleration engine. This subsystem is referred to as the TCP Offloading Engine (ToE).

The acceleration includes two functions:

- ToE (Transmit Segmentation Offload (TSO) and Large Reassembly Offload (LRO))
- Mcpy - See **Section 8.7**.

Figure 57 provides an overview of the components including in the storage application acceleration engine.

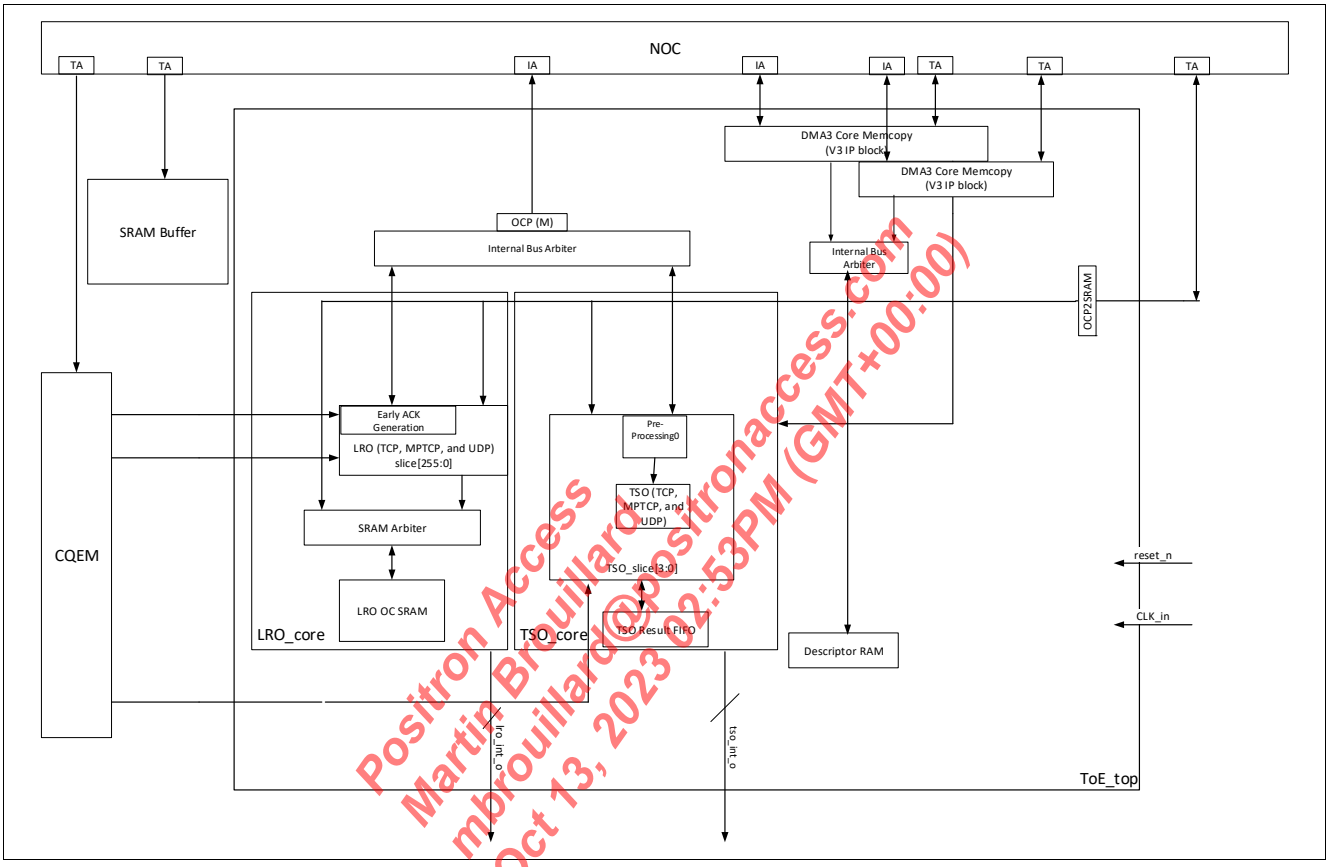


Figure 57 Storage Application Engine Overview

The storage application engine supports these features:

- Incorporates a pipeline-based hardware design to prevent stalls.
- Glitch-free system software interface for smooth integration and usage
- Supports Transmit Segmentation Offload (TSO) and Large Reassembly Offload (LRO) for TCP over IPv4/IPv6.
- Supports Generic Segmentation Offload (GSO) and Generic Reassembly Offload (GRO) for UDP over IPv4/IPv6.
- Supports MultiPath TCP (MTPCP) fragmentation and assembly.
- Supports segmentation of TCP/UDP for packets of up to 64 KB in size.
- Inserts L2, IP, and TCP headers for each segment.
- Four Transmit Segmentation Offload (TSO) hardware slices with 2 DMA memcpy engines
- 256 stall context for hardware and software synchronization
- 256 LRO ports
- TCP earlier ACK generation to speed up LRO flows
- Supports in-order Large Reassembly Offload (LRO).
- Gathers scattered data in the SKB buffer to pre-process for Transmit Segmentation Offload (TSO).
- Does not support IP tunneling.

8.6.1 TSO Function Description

Segmentation is performed without interrupting the CPU with the DMA memcpy function.

LRO is used for TCP packets. GRO is used for UDP packets. The function flow and software interface is unified.

8.6.1.1 Gathering of Scattered Data Pre-processing

The Linux kernel does not always store a large packet in a single memory location. Large packets are more likely broken up into fragments and stored in different, non-sequential memory locations.

The TSO uses the DMA memcpy function to gather these fragments and store them in one continuous memory location. These fragments may have a size of up to 64 KB and are located in the DDR memory.

The pointer to this memory location is specified in the Packet Buffer Pointer address field, provided by the TSO software command.

The the command gathering flag bit (G) indicates the meaning of the Packet Buffer Pointer field:

- When G is equal to 0, no data is gathered. This value is invalid and the pointer is not used.
- When G is equal to 1 for the first command set, and multiple command sets must be issued to the TSO to complete a gathering operation, the Packet Buffer Pointer field contains the pointer to the targeted large packet. For the second, and subsequent command sets, the field contains the pointer to the SKB buffer.

8.6.1.2 Segmentation Data Flow

The main steps of segmentation are:

1. The software sets up the BM for the TSO port and locates a buffer to store ToE segments.
2. The software pushes a TSO request command and starts the TSO engine.
3. The TSO engine reads in the request command data to get the user data window size and the MTU setting, and then begins segmentation.
4. The TSO inserts the modified IP length field and TCP sequence number into the newly formed segments.
5. The TSO enqueues the segments into the BM and the BM transfers those segments to their destination ports.

8.6.1.3 Segments Enqueue and BM Interface

When the segmentation process is finished, the TCP segments are sent to other interfaces via the BM enqueue port. This uses the standard BM DMA ports.

Every BM enqueue port has two descriptor register sets:

- A set for standard packet sizes up to 1.5 Kbytes, referred as Option 2. See **Table 109**.
- A set for jumbo packet sizes up to 9 Kbytes, referred as Option 3. See **Table 110**.

Note: The TSO only supports segment sizes of up to 4 Kbytes.

8.6.1.4 TSO Software Interface Description

This section describes the software interface used to access the TSO registers and the command structure format

The TSO uses the DMA memcpy function to move data. The setup of the DMA3 channel is performed through the DMA configuration port.

Configuration and control of the TSO engine, other than DMA3, is done through the NoC configuration path.

The CPU is able to program the ToE engine to perform segmentation on up to 64 packets at any one time, with one packet per session and 16 packets per port.

Note: Interleaved segmentation is not supported.

8.6.2 LRO Function Description

The LRO function does not physically assemble TCP segments into large packets, but rather checks the sequence numbers of the TCP segments, puts them in the right order, and passes the corresponding pointers to the software.

A dedicated BM LRO port provides two DMA descriptor entries.

LRO is used for TCP packets. GRO is used for UDP packets. The function flow and software interface is unified.

See **Section 8.6.4** for more information.

8.6.3 ToE Address Mapping and Channel Assignment

This section provides the internal ToE address map and DMA channel assignment.

Table 108 ToE Address Map

Start Address	End Address	Usage
ECB0 0000	ECB3 FFFF	TSO and LRO Register Space
ECB4 0600	ECB4 063F	Descriptor storage for DMA3a ch 4 and 5 for pre-process slice 0
ECB4 0700	ECB4 073F	Descriptor storage for DMA3b ch 4 and 5 for pre-process slice 0
ECB4 0800	ECB4 083F	Descriptor storage for DMA3a ch 6 and 7 for pre-process slice 1
ECB4 0900	ECB4 093F	Descriptor storage for DMA3b ch 6 and 7 for pre-process slice 1
ECB4 0A00	ECB4 0A3F	Descriptor storage for DMA3a ch 14 and 15 for pre-process slice 2
ECB4 0B00	ECB4 0B3F	Descriptor storage for DMA3b ch 14 and 15 for pre-process slice 2
ECB4 0C00	ECB4 0C3F	Descriptor storage for DMA3a ch 16 and 17 for pre-process slice 3
ECB4 0D00	ECB4 0D3F	Descriptor storage for DMA3b ch 16 and 17 for pre-process slice 3
ECB4 1000	ECB4 10FF	Descriptor storage for DMA3a ch 8
ECB4 1100	ECB4 11FF	Descriptor storage for DMA3a ch 9
ECB4 1200	ECB4 12FF	Descriptor storage for DMA3a ch 10
ECB4 1300	ECB4 13FF	Descriptor storage for DMA3a ch 11
ECB4 1400	ECB4 14FF	Descriptor storage for DMA3a ch 18
ECB4 1500	ECB4 15FF	Descriptor storage for DMA3a ch 19
ECB4 1600	ECB4 16FF	Descriptor storage for DMA3a ch 20
ECB4 1700	ECB4 17FF	Descriptor storage for DMA3a ch 21
ECB4 1800	ECB4 18FF	Descriptor storage for DMA3a ch 28
ECB4 1900	ECB4 19FF	Descriptor storage for DMA3a ch 29
ECB4 1A00	ECB4 1AFF	Descriptor storage for DMA3a ch 30
ECB4 1B00	ECB4 1BFF	Descriptor storage for DMA3a ch 31
ECB4 1C00	ECB4 1CFF	Descriptor storage for DMA3a ch 22
ECB4 1D00	ECB4 1DFF	Descriptor storage for DMA3a ch 23
ECB4 1E00	ECB4 1EFF	Descriptor storage for DMA3a ch 24
ECB4 1F00	ECB4 1FFF	Descriptor storage for DMA3a ch 25
ECB4 2000	ECB4 20FF	Descriptor storage for DMA3b ch 8
ECB4 2100	ECB4 21FF	Descriptor storage for DMA3b ch 9
ECB4 2200	ECB4 22FF	Descriptor storage for DMA3b ch 10
ECB4 2300	ECB4 23FF	Descriptor storage for DMA3b ch 11
ECB4 2400	ECB4 24FF	Descriptor storage for DMA3b ch 18
ECB4 2500	ECB4 25FF	Descriptor storage for DMA3b ch 19
ECB4 2600	ECB4 26FF	Descriptor storage for DMA3b ch 20
ECB4 2700	ECB4 27FF	Descriptor storage for DMA3b ch 21
ECB4 2800	ECB4 28FF	Descriptor storage for DMA3b ch 28
ECB4 2900	ECB4 29FF	Descriptor storage for DMA3b ch 29
ECB4 2A00	ECB4 2AFF	Descriptor storage for DMA3b ch 30

Table 108 ToE Address Map (cont'd)

Start Address	End Address	Usage
ECB4 2B00	ECB4 2BFF	Descriptor storage for DMA3b ch 31
ECB4 2C00	ECB4 2CFF	Descriptor storage for DMA3b ch 22
ECB4 2D00	ECB4 2DFF	Descriptor storage for DMA3b ch 23
ECB4 2E00	ECB4 2EFF	Descriptor storage for DMA3b ch 24
ECB4 2F00	ECB4 2FFF	Descriptor storage for DMA3b ch 25
ECB8 0000	ECB8 00A7	Primary result context storage for LRO port 0
ECB8 00A8	ECB8 014F	Secondary result context storage for LRO port 0
ECB8 0150	ECB8 01F7	Primary result context storage for LRO port 1
ECB8 01F8	ECB8 029F	Secondary result context storage for LRO port 1
.....	.....	.....
ECB9 4EB0	ECB9 4F54	Primary result context storage for LRO port 255
ECB9 4F58	ECB9 4FFF	Secondary result context storage for LRO port 255
ECB9 5000	ECB9 50FF	Nomatch result context storage for LRO ports
ECB9 5100	ECB9 70FF	Exception result context storage for LRO ports
ECB9 7100	ECB9 84FF	Stall context storage for LRO ports

LRO base address	0xECB8_0000	CMD Port 0		CMD Port 1		CMD Port 2		CMD Port 3		.....	CMD Port n		.....	CMD Port 254		CMD Port 255	
	offset per Port																
Result		0	00a8	150	01F8	2A0	348	3F0	498		DEC2HEX(84*4*n)	DEC2HEX(84*4*n+42*4)		14D60	14e08	14eb0	14f58
		4	AC	154	1FC	2A4	34C	3F4	49C					14D64	14E0C	14EB4	14F5C
		8	B0	158	200	2A8	350	3F8	4A0					14D68	14E10	14EB8	14F60
	1	C	B4	15C	204	2AC	354	3FC	4A4					14D6C	14E14	14EBC	14F64
		10	B8	160	208	2B0	358	400	4A8					14D70	14E18	14EC0	14F68
	2	14	BC	164	20C	2B4	35C	404	4AC					14D74	14E1C	14EC4	14F6C
		18	C0	168	210	2B8	360	408	4B0					14D78	14E20	14EC8	14F70
	3	1C	C4	16C	214	2BC	364	40C	4B4					14D7C	14E24	14ECC	14F74
		20	C8	170	218	2C0	368	410	4B8					14D80	14E28	14ED0	14F78
	4	24	CC	174	21C	2C4	36C	414	4BC					14D84	14E2C	14ED4	14F7C
		28	D0	178	220	2C8	370	418	4C0					14D88	14E30	14ED8	14F80
	5	2C	D4	17C	224	2CC	374	41C	4C4					14D8C	14E34	14EDC	14F84
		30	D8	180	228	2D0	378	420	4C8					14D90	14E38	14EE0	14F88
	6	34	DC	184	22C	2D4	37C	424	4CC					14D94	14E3C	14EE4	14F8C
		38	E0	188	230	2D8	380	428	4D0					14D98	14E40	14EE8	14F90
	7	3C	E4	18C	234	2DC	384	42C	4D4					14D9C	14E44	14EEC	14F94
		40	E8	190	238	2E0	388	430	4D8					14DA0	14E48	14EF0	14F98
	8	44	EC	194	23C	2E4	38C	434	4DC					14DA4	14E4C	14EF4	14F9C
		48	F0	198	240	2E8	390	438	4E0					14DA8	14E50	14EF8	14FA0
	9	4C	F4	19C	244	2EC	394	43C	4E4					14DAC	14E54	14EFC	14FA4
		50	F8	1A0	248	2F0	398	440	4E8					14DB0	14E58	14F00	14FA8
	10	54	FC	1A4	24C	2F4	39C	444	4EC					14DB4	14E5C	14F04	14FAC
		58	100	1A8	250	2F8	3A0	448	4F0					14DB8	14E60	14F08	14FB0
	11	5C	104	1AC	254	2FC	3A4	44C	4F4					14DBC	14E64	14F0C	14FB4
		60	108	1B0	258	300	3A8	450	4F8					14DC0	14E68	14F10	14FB8
	12	64	10C	1B4	25C	304	3AC	454	4FC					14DC4	14E6C	14F14	14FBC
		68	110	1B8	260	308	3B0	458	500					14DC8	14E70	14F18	14FC0
	13	6C	114	1BC	264	30C	3B4	45C	504					14DCC	14E74	14F1C	14FC4
		70	118	1C0	268	310	3B8	460	508					14DD0	14E78	14F20	14FC8
	14	74	11C	1C4	26C	314	3BC	464	50C					14DD4	14E7C	14F24	14FCC
		78	120	1C8	270	318	3C0	468	510					14DD8	14E80	14F28	14FD0
	15	7C	124	1CC	274	31C	3C4	46C	514					14DDC	14E84	14F2C	14FD4
		80	128	1D0	278	320	3C8	470	518					14DE0	14E88	14F30	14FD8
	16	84	12C	1D4	27C	324	3CC	474	51C					14DE4	14E8C	14F34	14FDC
		88	130	1D8	280	328	3D0	478	520					14DE8	14E90	14F38	14FE0
	17	8C	134	1DC	284	32C	3D4	47C	524					14DEC	14E94	14F3C	14FE4
		90	138	1E0	288	330	3D8	480	528					14DF0	14E98	14F40	14FE8
	18	94	13C	1E4	28C	334	3DC	484	52C					14DF4	14E9C	14F44	14FEC
		98	140	1E8	290	338	3E0	488	530					14DF8	14EA0	14F48	14FF0
	19	9C	144	1EC	294	33C	3E4	48C	534					14DFC	14EA4	14F4C	14FF4
		A0	148	1F0	298	340	3E8	490	538					14E00	14EA8	14F50	14FF8
	20	A4	14C	1F4	29C	344	3EC	494	53C					14E04	14EAC	14F54	14FFC

Figure 58 LRO OC Detailed Address

ToE and memcpy share one DMAm instance.

Table 109 and Table 110 show the channel number to module assignment, along with its function.

Table 109 DMAm (DMA3a,b) Hardware Stage - Two DMA Core - Option 2

Channel Number		TSO Module Activity	Function	DMA Number
Rx	Tx			
TSO Slice 0				
4	5	Pre-processing	Gather fragments to form a single large packet	DMA3a
0	1	Segment 0	–	–
2	3	Segment 1	–	–
8	9	Enqueue segment 0	Copy the first segmented packet from SRAM to DDR	DMA3b
10	11	Enqueue segment 1	Copy the second segmented packet from SRAM to DDR	DMA3b
TSO Slice 1				
6	7	Pre-processing	Gather fragments to form a single large packet	DMA3a
18	19	Enqueue segment 0	Copy the first segmented packet from SRAM to DDR	DMA3b
20	21	Enqueue segment 1	Copy the second segmented packet from SRAM to DDR	DMA3b
TSO Slice 2				
14	15	Pre-processing	Gather fragments to form a single large packet	DMA3a
28	29	Enqueue segment 0	Copy the first segmented packet from SRAM to DDR	DMA3b
30	31	Enqueue segment 1	Copy the second segmented packet from SRAM to DDR	DMA3b
TSO Slice 3				
16	17	Pre-processing	Gather fragments to form a single large packet	DMA3a
22	23	Enqueue segment 0	Copy the first segmented packet from SRAM to DDR	DMA3b
24	25	Enqueue segment 1	Copy the second segmented packet from SRAM to DDR	DMA3b

Table 110 DMAm (DMA3a,b) Hardware Stage - Two DMA Core - Option 3

Channel Number		TSO Module Activity	Function	DMA Number
Rx	Tx			
TSO Slice 0				
4	5	Pre-processing	Gather fragments to form a single large packet	DMA3a,3b
0	1	Segment 0	–	–
2	3	Segment 1	–	–
8	9	Enqueue segment 0	Copy the first segmented packet from SRAM to DDR	DMA3a,3b
10	11	Enqueue segment 1	Copy the second segmented packet from SRAM to DDR	DMA3a,3b
TSO Slice 1				
6	7	Pre-processing	Gather fragments to form a single large packet	DMA3a,3b
18	19	Enqueue segment 0	Copy the first segmented packet from SRAM to DDR	DMA3a,3b
20	21	Enqueue segment 1	Copy the second segmented packet from SRAM to DDR	DMA3a,3b
TSO Slice 2				
14	15	Pre-processing	Gather fragments to form a single large packet	DMA3a,3b
28	29	Enqueue segment 0	Copy the first segmented packet from SRAM to DDR	DMA3a,3b
30	31	Enqueue segment 1	Copy the second segmented packet from SRAM to DDR	DMA3a,3b
TSO Slice 3				
16	17	Pre-processing	Gather fragments to form a single large packet	DMA3a,3b
22	23	Enqueue segment 0	Copy the first segmented packet from SRAM to DDR	DMA3a,3b
24	25	Enqueue segment 1	Copy the second segmented packet from SRAM to DDR	DMA3a,3b

8.6.4 LRO Software Interface Description

The software interface includes register access, as well as result context format.

Since LRO does not move the data, it is unnecessary to involve DMA. The PMAC header and TCP header are read in by the OCP master without DMA.

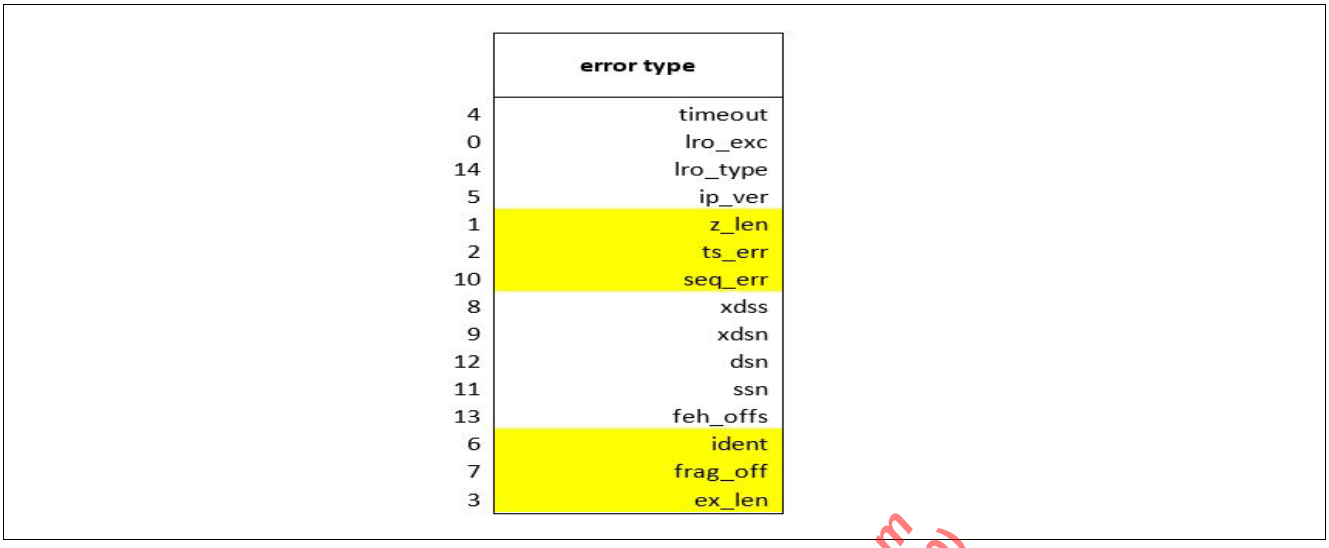
Use the NoC configuration path to configure and control the LRO engine. The CPU programs the command register and LRO takes the information.

The CPU is able to program up to 256 sessions (one session per port) for the LRO engine to process at one time.

The LRO result is written into the result context which the LRO informs the CPU to fetch by interrupt indication. Since each port has two result contexts, each context port generates one interrupt, each result context written triggers the corresponding interrupt. A context indication bit indicates which context (0 or 1) is written by hardware for this interrupt.

Definition of Flags

- Bit 0 is set to 1 when any aggregation is performed. It is set to 0 for a single packet session.
- Bit 1 is set to 1 when performing the TCP timestamp check. Otherwise, it is set to 0.
- Bit 2 is set to 1 when handling an exception case. Otherwise, it is set to 0.
- Bit [6:3] is used for an exception case encoding. **Figure 59** provides the complete list.
  - 0000: LRO exception detected from the packet processor
  - 0001: Zero length packet
  - 0010: TCP timestamp check error
  - 0011: Excessive length
  - 0100: Session timeout. These are potential use cases:
    Timeout happens while the LRO is processing a packet.
    Timeout happens while the LRO is idle, there is a new incoming packet from CQM and an OC aggregation is ongoing.
    Timeout happens while the LRO is idle, there is no new incoming packet from CQM and there is an OC aggregation ongoing.
  - 0101: Unknown IP version
  - 0110: Identifier mismatch
  - 0111: Frag off, unexpected fragment offset
  - 1000: Non-DSS MPTCP sub type
  - 1001: No DSN present
  - 1010: Sequence error
  - 1011: SSN mismath
  - 1100: DSN mismatch
  - 1101: Frag ext header offset is 0
  - 1110: LRO type mismatch between the descriptor and the command
- Bit [14:10]: Number of segments collected
- Bit [30:15]: Length of segments collected



	error type
4	timeout
0	lro_exc
14	lro_type
5	ip_ver
1	z_len
2	ts_err
10	seq_err
8	xdss
9	xdsn
12	dsn
11	ssn
13	feh_offs
6	ident
7	frag_off
3	ex_len

Figure 59 LRO Error Check Order

Definition of Exception Entry Flags

- Unmatched entry flag: bit [31:16], total 16 mismatch entries
- Exception entry flag:16 x bit [31:0], total 256 entries, one port one entry, two bits per entry

Programming Hints

Table 111 lists questions and answers software programmers may have while using LRO.

Table 111 Q&A Regarding Use of LRO

Questions	Answers
When the LRO threshold/MaxAggr is reached, the result goes to the OC and an interrupt is raised?	Yes
Will there be no exception in this case?	Yes
Will partial result be written to the OC and an interrupt raised even before this happens and when no exception has happened?	No
Can I get an OC interrupt before MaxAggr is reached and when no exception has happened?	No
After MaxAggr is reached, and the result is written to the OC, is a session stop/start required from the driver?	A session stop/start is not required.
When an exception condition occurs, is this understanding correct?	
• The partially aggregated packet result is written to the OC, and an OC interrupt is raised.	Yes
• The exception packet is written to the exception context and an exception context interrupt is raised.	Yes. Only the last exception packet is written to the exception context.
• The hardware expects the LRO driver to stop and start the same LRO flow session.	A session stop/start is not required.
• When a packet comes with a FID not programmed to the LRO engine, this only goes to the nomatch context.	Yes
• In case of unmatched context overflow, the LRO stalls and the packets are not sent to the stall context.	Yes

8.7 Memory Copy DMA Acceleration

This section describes the functionality of the memcpy (Mcpy) acceleration hardware.

The module is reused from GRX550 Series which used MIPS, however the virtual to physical address translation logic is updated to support the Intel Atom CPU.

8.7.1 Coherent Memory Copy Engine

The DMA memcpy function is extended for data copy engine with additional logic.

The system software provides the source address, destination address and amount of data to be moved. The memory copy engine proceeds with the operation and returns the result without additional system involvement.

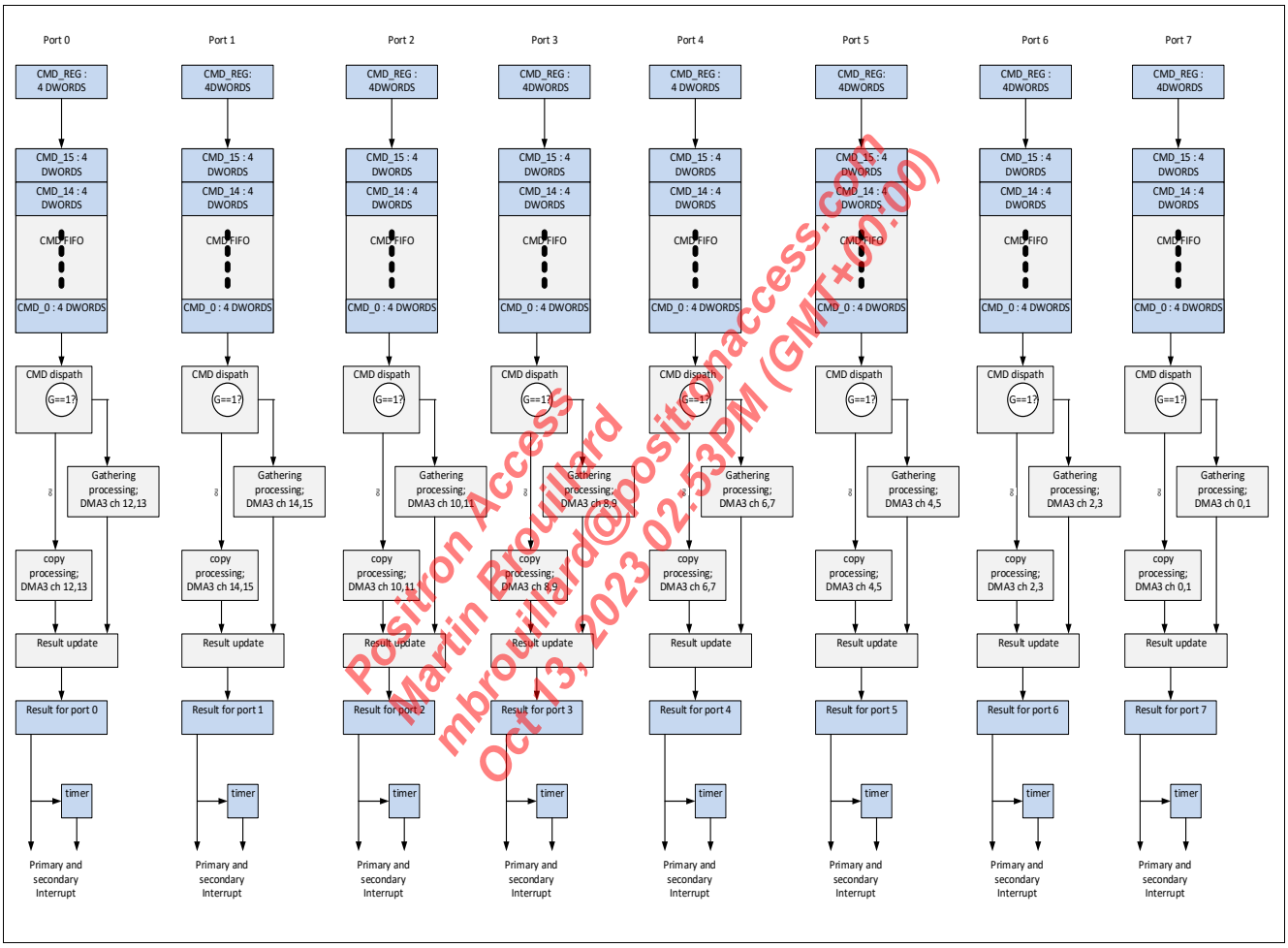


Figure 60 Coherent Memory Copy Engine Slices View

8.7.1.1 Features

These features are supported:

- Handle unaligned accesses (not on DWORD boundary)
  - [Header] + Aligned burst + [Footer]
  - To avoid extra checks, must be incorporated in the `memcpy()` routine.
- Multiple request queues per CPU core
  - At least a high priority queue and low priority queue per core/VM
  - Assure independence of individual VMs and their memcpy requests.
  - With four VMs, there are eight ports to accept request commands.
- Multi-request handling
  The memcpy block handles chunks of a transfer from a single large request at the same time as it handles other requests from other queues. It allows not to block all the queues while the first transfer is complete.
  - For example, a 1 MB memcpy should not be blocking all the other request queues.
- Data gathering mode
  By giving multiple data pointer points to scatter data blocks in the memory, the Mcpy performs gather copy to form a large continuous data block.
- Address translation from CPU virtual/linear to physical
  - 32 bit linear address, 4 KB page size
  - 32e (36/40/64bit) linear address, 4 KB and 2 MB page size
  - Per Mcpy port one CR3 address register

9 Interfaces and System Functions

This chapter describes these sections:

9.1 Universal Asynchronous Receiver/Transmitter

There are four Asynchronous Serial Channel (ASC) modules integrated in the chip. Each of the ASC module is primarily used as UART. **Figure 61** and **Figure 62** show the ASC module configuration.

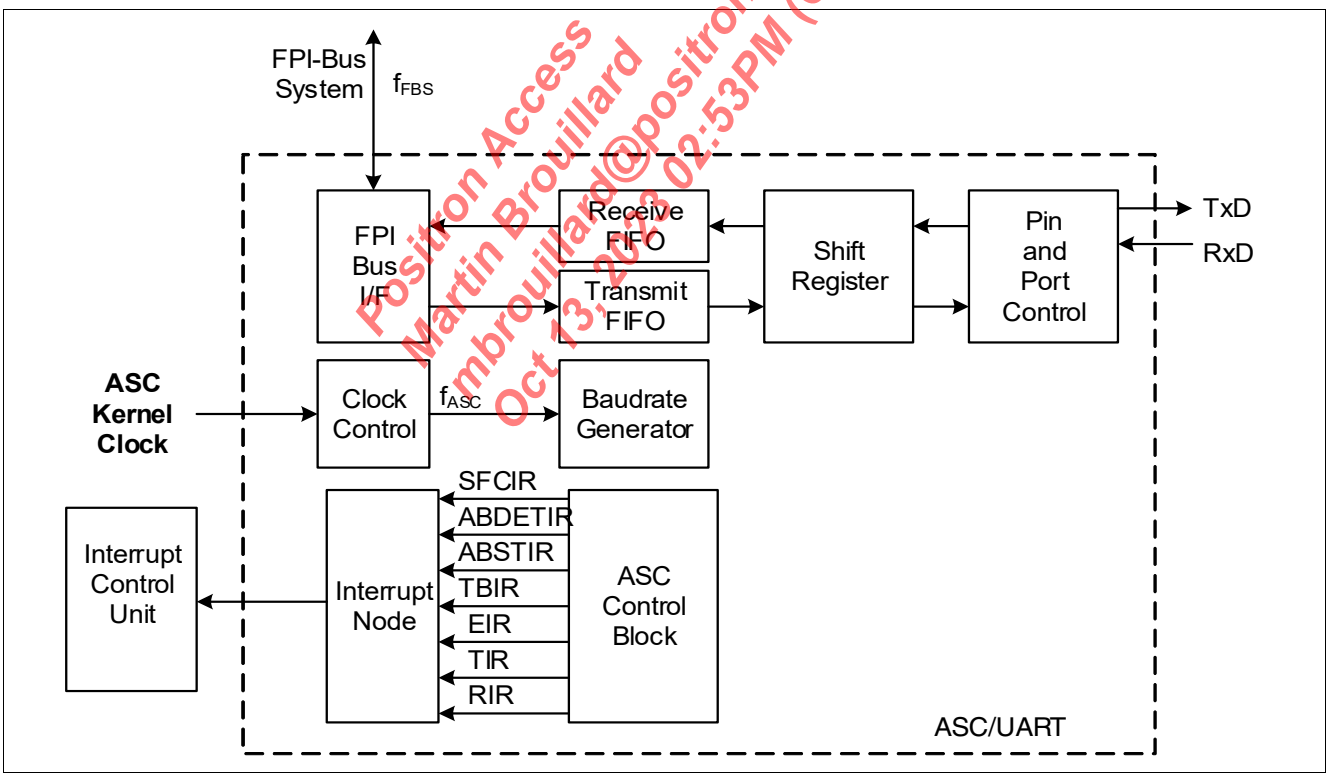


Figure 61 ASC/UART0,1 Block Diagram

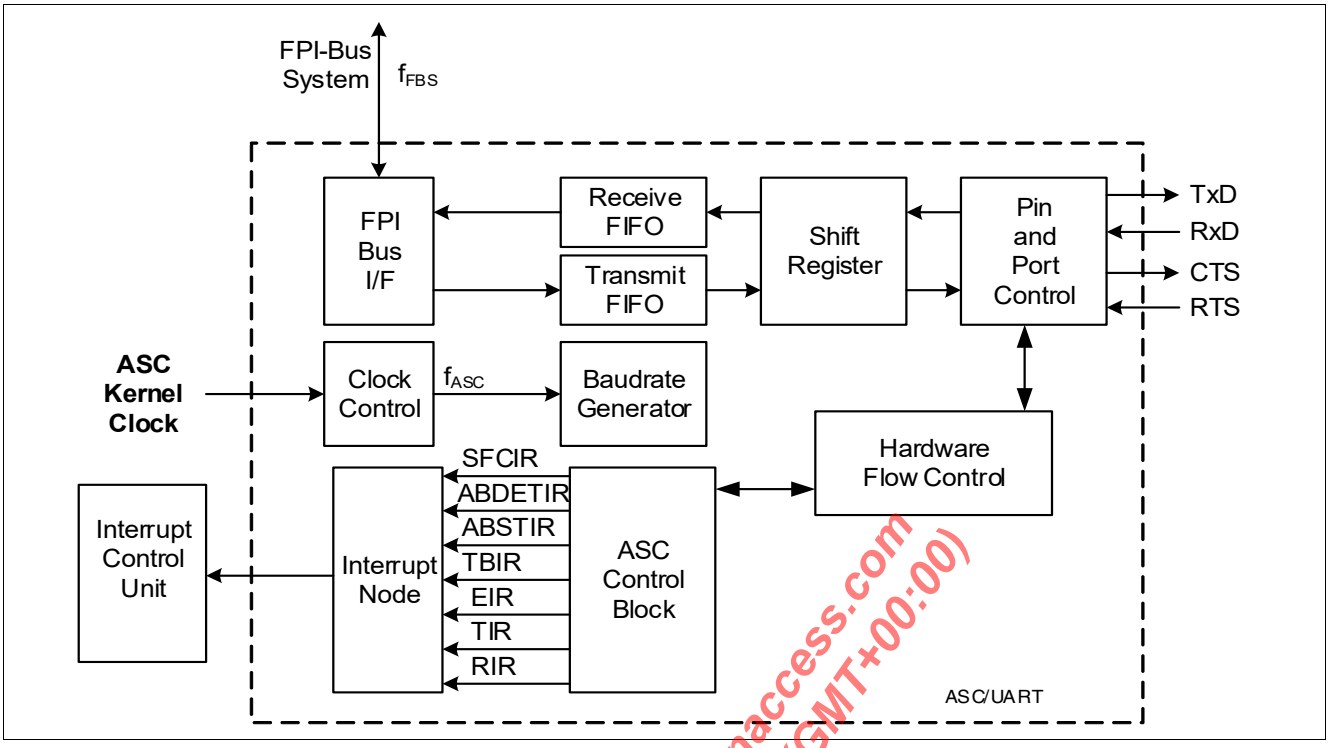


Figure 62 ASC/UART2,3 Block Diagram

9.1.1 ASC Features

The ASC module provides these features:

- ASC2 and ASC3 provided hardware-based flow control (CTS and RTS pin)
- Automatic clock shut on/off, depending on line and/or bus activity
- Full-duplex asynchronous operating modes
  - 7-bit or 8-bit data frames, LSB first
  - Parity bit generation/checking
  - One or two stop bits
  - Baud rate from 12.6 MBaud to 426 Baud at 200 MHz module fixed clock
- Programmable XON/XOFF flow control
  - Programmable XON and XOFF characters
  - Programmable automatic flow control. XON/XOFF characters are automatically inserted into/removed from the data stream, depending on the FIFO status.
- Line break detection
- Programmable end-of-message detection based on character matching or time-out
- Multiprocessor mode for automatic address/data byte detection
- Loop-back capability
- Double buffered transmitter/receiver
- Interrupt generation
  - On a transmitter buffer empty condition
  - On a transmit last bit of a frame condition
  - On a receiver buffer full condition
  - On an error condition (frame, parity, overrun error, line break)
  - On XOFF reception on modem control/status signal changes
- Autobaud detection unit for asynchronous operating modes
  - Detection of standard baud rates
    1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, and 230400 Baud

  - Detection of non-standard baud rates
  - Detection of asynchronous modes
  - 7-bit, even parity; 7-bit, odd parity
    8-bit, even parity; 8-bit, odd parity; 8-bit, no parity automatic initialization of control bits and baud rate generator after detection
  - Detection of a serial two-byte ASCII character frame
- FIFO
  - 16 X 8 receive FIFO (RxFIFO), 8-bit organized for ASC
  - 16 X 8 transmit FIFO (TxFIFO), 8-bit organized for ASC
  - Independent control of RxFIFO and TxFIFO depth and width
  - 8-bit FIFO data width without segmentation or reassembly in 8-bit configuration
  - Programmable receive/transmit interrupt trigger level
  - Receive and transmit FIFO filling level indication
  - Overrun error generation
  - Underflow error generation

9.1.2 General Operation

The ASC supports full-duplex asynchronous communication up to 12.6 MBaud at 200 MHz module clock. Either 7-bit or 8-bit data transfer, parity generation, and the number of stop bits is selectable. Parity, framing, and overrun error detection is provided to increase the reliability of data transfers. Transmission and reception of data is double-buffered. For multiprocessor communication, a mechanism is provided to distinguish address bytes from data bytes. A 13-bit baud rate timer with a versatile input clock divider circuitry provides the serial clock signal. In a special asynchronous mode, the ASC autobaud detection allows to detect asynchronous data frames with its baud rate and mode with automatic initialization of the baud rate generator and the mode control bits.

A transmission is started by writing to the transmit buffer register TBUF. In a 32-bit configuration of the transmit path, a segmentation module breaks the 32-bit word into 4 8-bit characters to be transmitted. The selected operating mode determines the number of data bits that are actually transmitted, so that bits written to position 7, 15, 23, and 31 are ignored in all modes with 7 bits of data only. Data transmission is double-buffered, so a new character may be written to the transmit buffer register before the transmission of the previous character is complete. This allows the transmission of characters back-to-back without gaps.

Data reception is enabled by activating the receiver enable bit STATE.REN via a write to WHBSTATE.SETREN. In the 32-bit configuration, a reassembly module combines 8-bit characters received to 32-bit words to be put into the FIFO and the RBUF register. Bits at the positions 7, 15, 23, and 31 of RBUF not valid in the selected operating mode are read as zeros. After completing the reassembly of a word, the received data is readable from the (read-only) receive buffer register RBUF.

Data reception is double-buffered, so that reception of a second character may already begin before the previously received character is read out of the receive buffer register. In all modes, receive overrun error detection is selectable through bit MCON.ROEN. When enabled, the overrun error status flag STATE.ROE and the error interrupt request line EIR are activated when the receive buffer register is not read by the time the reception of a ninth character is complete. The previously received character in the receive buffer is overwritten.

The operating mode of the serial channel ASC is controlled by its control register MCON. This register contains control bits for mode and error check selection. Status flags for error identification are found in the STATE register.

Attention: The ASC supports up to a 12.6M baud-rate in the serial interface, however the overall achievable system level performance depends also on how fast the host is able to feed data into/take out data from the ASC's FIFO.

9.1.2.1 Baud Rate Generator

For all asynchronous modes, the baud rate generator provides a clock f_{BRT} with 16 times the rate of the established baud rate. Every received bit is sampled at the 7th, 8th, and 9th cycle of this clock. The clock divider circuitry generating the input clock for the 13-bit baud rate timer is extended by a fractional divider circuitry allowing adjustment for a more accurate baud rate and the extension of the baud rate range.

The baud rate of the baud rate generator depends on these bits and register values:

- Input clock f_{hw_clk}
- Selection of the baud rate timer input clock `fDIV` by bits `MCON.FDE` and `MCON.BRS`
- When the `MCON.FDE` bit is set (fractional divider): value of the `FDV.FD_VAL` register
- Value of the 13-bit reload register Baudrate Timer/Reload Register (BG)

The output clock of the baud rate timer with the reload register is the sample clock in the ASC asynchronous modes. For baud rate calculations, this baud rate clock `fBR` is derived from the sample clock `fDIV` divided by 16.

The ASC Fractional Divider Register (FDV) contains the 9-bit divider value for the fractional divider. It is also used for reference clock generation of the autobaud detection unit.

Using the Fixed Input Clock Divider

Table 112 shows the formulas that determine the baud rate when using the fixed input clock divider ratios (MCON.FDE=0) and the required reload value for a given baud rate.

Table 112 Baud Rate Formulas Using the Fixed Input Clock Dividers

FDE	BRS	BG	Formula
0	0	0... 8191	$\text{Baudrate} = \frac{f_{ASC}}{32 \times (BG + 1)}$ F1
	1		$\text{Baudrate} = \frac{f_{ASC}}{48 \times (BG + 1)}$ F3

BG represents the contents of the reload register `BG.BR_VAL`, taken as unsigned 13-bit integer.

The maximum baud rate achievable when using the two fixed clock divider and a module clock of 84 MHz is 2.625 MBaud. Table 113 lists various commonly used baud rates together with the required reload values and the deviation errors compared to the intended baud rate.

Table 113 Typical Asynchronous Baud Rates Using the Fixed Input Clock Dividers

Baud Rate	MCON.BRS = '0', f_{hw_clk} = 200 MHz		MCON.BRS = '1', f_{hw_clk} = 200 MHz	
	Deviation Error	Reload Value	Deviation Error	Reload Value
6.25 MBaud	---	0000_H	NA	NA
19.2 KBaud	+0.16 % / -0.15 %	00144_H / 00145_H	+0.1 % / -0.45 %	$00D8_H$ / $00D9_H$

Note: `MCON.FDE` must be 0 to achieve the baud rates listed in Table 113.
The provided deviation errors are rounded.

Using the Fractional Divider

When the fractional divider is selected, the input clock f_{DIV} for the baud rate timer is derived from the module clock f_{hw_clk} by a programmable divider. When `MCON.FDE` is set, the fractional divider is activated. It divides f_{hw_clk} by a fraction of n/512 for any value of n from 0 to 511. When n=0, the divider ratio is 1, which means that $f_{DIV} = f_{hw\_clk}$.

Note: The highest possible baud rate is achieved by enabling the fractional divider and setting the divider value to 0. With BG set to 0 as well, the achieved baud rate is 1/16 of the module clock f_{hw_clk}. Higher baud rates cannot be achieved due to the 16 fold oversampling of the incoming data.

In general, the fractional divider allows the baud rate to be programmed with much more accuracy than with the two fixed prescaler divider stages.

Table 114 Baud Rate Formulas Using the Fractional Input Clock Divider

FDE	MCON.BRS	BG	FDV	Formula
1	-	0... 8191	1... 511	$\text{Baudrate} = \frac{\text{FDV}}{512} \times \frac{f_{ASC}}{16\ (BG+1)}$ (1)
			0	$\text{Baudrate} = \frac{f_{ASC}}{16 \times (BG + 1)}$ F5 (2)

Notes

1. *BG represents the contents of the reload register `BG.BR_VAL`, taken as an unsigned 13-bit integer.*
2. *FDV represents the contents of the fractional divider register `FDV.FD_VAL` taken as an unsigned 9-bit integer.*

Table 115 Typical Baud Rates Using the Fractional Input Clock Divider

F_{hw_clk}	Desired Baud Rate	BG	FDV	Resulting Baud Rate	Deviation
200 MHz	115.2 kBaud	66_H	$1E6_H$	115.196 kBaud	0.003 %
	57.6 kBaud	66_H	$F3_H$	57.598 kBaud	0.003 %
	38.4 kBaud	$4C_H$	79_H	38.365 kBaud	0.09 %

9.2 Serial Peripheral Interfaces

The Synchronous Serial Communication (SSC) module supports industry compliant Serial Peripheral Interface (SPI) interface. There are four SSC modules integrated in the SoC.
The SPI is equipped with a receive as well as a transmit FIFO with eight 32-bit wide entries each.

9.2.1 Features

The SPI supports:

- Master and slave mode operation
  - Full-duplex or half-duplex operation
  - SPI instance #0 supports either master or slave mode
  - SPI instance #1, 2, and 3 supports only master mode
  - SPI0: 3 chip select; master and slave mode
  - SPI1: 6 chip select; master mode only
  - SPI2: 3 chip select; master mode only
  - SPI3: 2 chip select; master mode only
- Flexible data format
  - Programmable number of data bits per SSC-word: 2 to 32 bits
  - Programmable shift direction: LSB or MSB shift first
  - Programmable clock polarity: idle low or high state for the shift clock
- Programmable clock/data phase: data shift with leading or trailing edge of the shift clock SPI-compatible
  - Data and clock polarity in all SPI modes
  - Programmable idle period after address and between data
- Baud rate generation from a maximum of 50 MHz down to (200/131072) MHz
- Interrupt generation
  - On a transmitter empty condition
  - On a receiver full condition
  - On an error condition (receive, transmit, abort, or mode error)
- Interfaces
  - FPI interface for CPU accesses
  - DMA0 Tx and Rx channel connected
  - CPU accesses via FPI take precedence
- FIFO
  - Up to 64 SSC-words receive FIFO (RxFIFO)
  - Up to 64 SSC-words transmit FIFO (TxFIFO)
  - Independent control of RXFIFO and TxFIFO
  - 32-bit FIFO data width
  - Programmable receive/transmit interrupt trigger Level
  - Receive and transmit FIFO filling level indication
  - Overrun error generation
  - Underflow error generation

9.2.2 Functional Description

The high-speed SPI supports both full-duplex and half-duplex serial synchronous communication up to 1/2 of the module clock frequency in master mode or 1/4 of the module clock frequency in slave mode and down to 1/131072 of the module clock frequency. The serial clock signal is either generated by the SPI itself (master mode) or received from an external master (slave mode). Data width, shift direction, clock polarity, and phase are programmable. Up to eight chip select outputs are generated by the master. A chip select input can be used to turn off the device in slave mode. The SPI is compatible with the Motorola SPI interface. Data transmission and reception is double-buffered. A 16-bit baud rate generator provides the SPI with a separate serial clock signal.

9.2.2.1 General Operation

The SPI supports full-duplex and half-duplex synchronous communication up to 1/2 of the module clock in master mode and 1/4 of the module clock in slave mode. The module clock f_{ssc_clk} itself is at most the same as the bus clock f_{fpi_clk}, but may be scaled down for power savings by a clock divider controlled by the CLC register. The serial clock signal can be generated by the SPI itself (master mode) or can be received from an external master (slave mode).

Data width, shift direction, clock polarity, and phase are programmable. The data width of an SSC-word is programmable in bits, not bytes. The remaining bits in the transmit or receive buffer are treated as *don't care*. Up to eight chip select outputs can be generated by the master. A chip select input can be used to turn off the device in slave mode. The SPI is compatible with Motorola SPI interface. Transmission and reception of data is double-buffered. A 16-bit baud rate generator provides the SPI with a separate serial clock signal.

The high-speed synchronous serial interface can be configured in a very flexible way, so it can be used with other synchronous serial interfaces, can serve for master/slave or multi master interconnections or can operate compatible with the popular SPI interface. Thus, the SPI can be used to communicate with shift registers (IO expansion), peripherals (for example EEPROMs) or other controllers (networking).

The SPI supports half-duplex and full-duplex communication. Data is transmitted or received on lines TXD and RXD, connected with the Master Transmit/Slave Receive `MTSR` and Master Receive/Slave Transmit `MRST` pads. The clock signal is output via the Master Serial Shift Clock `MS_CLK` line or input via the Slave Serial Shift Clock `SS_CLK` line. Both clock lines are connected to Serial Shift Clock `SCLK`. These pins are alternate functions of port pins.

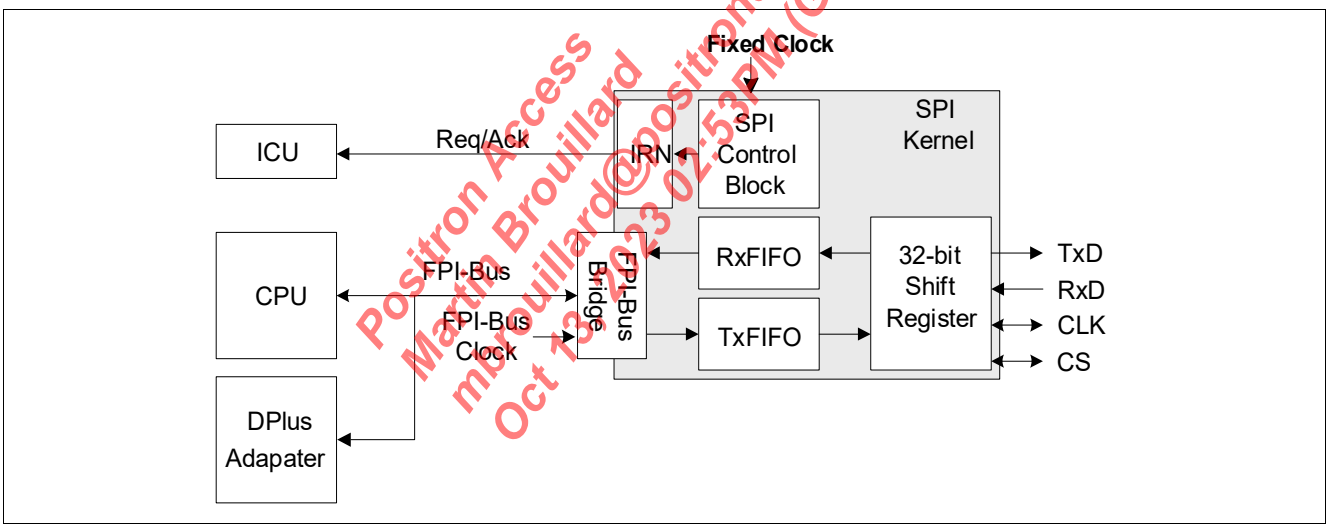


Figure 63 Synchronous Serial Channel SPI Block Diagram

9.2.2.2 Operating Mode Selection

The operating mode of the serial channel SPI is controlled by its `CON` and `WHBSTATE` control registers. The current operating mode is checked by the `STATE` status register. In programming mode where SPI disabled by `STATE.EN=0`, the `CON` register provides access to a set of control bits. In operation mode where SPI enabled by `STATE.EN=1`, the `STATE` register provides access to a set of status flags. To switch between operation modes, enabled/disabled and master/slave, use the `WHBSTATE` register.

Changing the operation mode must not be done while the SPI is busy with a serial transmission. The software must wait until all data in the transmit buffer/FIFO is sent out, then read all the received words to ensure that none are lost. Otherwise, the software can flush the transmit FIFO to cut the transmission short to the word currently under transmission and read or flush the receive FIFO. It has to wait until the current running transmission terminates and read/flush the simultaneously received word. The `STATE.BSY` flag is low and a reconfiguration can safely be started.

The shift register of the SPI is connected to both the transmit lines and the receive lines via the pin control logic. Transmission and reception of serial data are synchronized and take place at the same time. The same number of transmitted bits is also received. When simultaneous transmission and reception is not desired, the RXOFF and TXOFF bits in the CON register allow to respectively switch off the receiver or transmitter.

Transmit-only

When CON.RXOFF = 1, the receiver is turned off, the value on the input data line is *don't care* and not fed into the receiver buffer or FIFO. As a consequence, no receive interrupts occur.

Receive-only

When CON.TXOFF = 1 in master mode, the transmitter is turned off and no serial clock is generated until register RXREQ.RXCNT is written with the number of bytes to be read.

The data on the serial line must be some multiple of bytes, otherwise garbage is received. The 32-bit SSC-words are assembled from the received bytes. The number of shift clocks that are generated is RXREQ.RXCNT times 8. The transmit data output is set to the idle value during that time. When the number of bytes in RXREQ.RXCNT is not divisible by 4, the last 32-bit word is put into the receive buffer/FIFO as incomplete 32-bit word with the number of valid bytes specified in STATE.RXBV. In any case, the last (incomplete) word that is received is marked as end-of-message, which is important to the DMA interface. With the reception of the last byte of the receive request, due to the end-of-message condition, a receive interrupt is generated regardless whether the receive FIFO is filled to the programmed limit or not.

The status register RXCNT.TODO allows to check, how many data words still have to be received. As long as the transmitter is in master mode, any write operation to RXREQ.RXCNT has no effect, except changing this register. Even when the transmitter is turned off later, no receive operation is started unless RXREQ.RXCNT is written anew.

While the transmitter is turned off, any data that still is in the transmit FIFO or is written to the transmit FIFO anew remains there unchanged. Interrupts related to the transmit FIFO occur only in reaction to write transactions into the transmit FIFO, but not because data is removed from the transmit FIFO, since this data is not used for the serial transmission.

When the transmitter is turned off in slave mode, idle data values are sent to the master for every serial bit requested by the master's serial clock. Write operations to register RXREQ.RXCNT have no effect, except changing the register value.

Transmit and Receive

Transmit data is written into the Transmit Buffer (TB).

The busy flag STATE.BSY is set as soon as data is in the transmit buffer or FIFO and remains as long as the serial transmission lasts. The data word is moved to the shift register as soon as this is empty. The Transmit Interrupt Request (TIR) line is activated to indicate that register TB may be reloaded again.

When a transmit FIFO is implemented, TIR is only activated when the FIFO fill status is below the programmed threshold after moving a word to the shift register. TIR is used as a notifier to prevent transmit buffer underrun conditions.

An SPI master (STATE.MS=1) immediately begins transmitting when data is in the transmit buffer (or transmit FIFO), while an SPI slave (STATE.MS=0) waits for an active shift clock. When the programmed number of bits is transferred, the contents of the receive shift register are moved to the Receive Buffer (RB) and the Receive Interrupt Request (RIR) line is activated. The number of bits for an SSC-word can be programmed as 2..32 in CON.BM, when CON.ENBV = 0, or derived from the valid number of bytes from the FPI opcode/DMA byte enable vector, when CON.ENBV = 1.

When a receive FIFO is implemented, RIR is activated only when the FIFO fill status is higher than the programmed threshold. RIR is used as a notifier to prevent receive buffer overrun conditions.

When the TB is empty, no further transfer takes place. STATE.BSY is cleared when the previous word is transmitted while a word is simultaneously received and the received word is read or flushed from the receive buffer. The software cannot modify STATE.BSY, as this flag is hardware-controlled.

Note: Only one master can be on the SPI bus at a given time.

The SPI protocol defines four different modes, which are directly related to the four possible combinations of the CON.PO and CON.PH settings. **Table 116** lists this relationship.

Table 116 SPI Modes

SPI mode	CON.PO	CON.PH
0	0	1
1	0	0
2	1	1
3	1	0

SPI modes 0 and 3 are the most common in serial flash devices. In both modes, data is latched in at the rising edge of the serial clock.

9.2.2.3 Baud Rate Generation

The serial channel SPI has its own dedicated 16-bit baud rate generator with 16-bit reload capability, allowing baud rate generation independent of the timers.

The baud rate generator is clocked with the module clock f_{ssc_clk}. The timer counts downwards. Register BR is the baud rate reload register, which can be programmed and read. Register BRSTAT gives access to the baud rate counter register. Reading BRSTAT, while the SPI is enabled, returns the content of the timer. Reading BRSTAT, while the SPI is disabled, returns the former baud rate counter value from the last time the SPI was active. The desired reload value can be written to BR while the SPI is disabled only.

Note: Never write to BR while the SPI is enabled. The programmed change will be lost.

Figure 64 shows the calculation of either the resulting baud rate for a given reload value, or the required reload value for a given baud rate.

$$\text{Baudrate} = \frac{f_{SSC\_CLK}}{2 * (
 + 1)} \qquad BR = \frac{f_{SSC\_CLK}}{2 * \text{Baudrate}} - 1$$

Figure 64 Baud Rate

 represents the contents of the reload register, taken as unsigned 16-bit integer; while Baudrate is equal to f_{MS_CLK/SS_CLK}. The maximum achievable baud rate when using a module clock of 200 MHz is 50 MBaud (with
 = 0001_H).

Table 117 lists some possible baud rates together with the required reload values and the resulting baud rate, assuming a module clock of 200 MHz.

Table 117 Typical Baud Rates of the SPI (f_{ssc_clk} = 200 MHz)

Reload Value	Baud Rate (= f_{MS_CLK/SS_CLK})
0001_H	50 MBaud
0003_H	25 MBaud
0009_H	20 MBaud

9.3 I²C Controller

There are four I²C master controller (I²C) integrated in chip.

9.3.1 I²C Functional Overview

The Inter-IC (I²C) bus was developed by Philips Semiconductors. This specification is compliant with the I²C-version 2.1.

The I²C is a simple, bi-directional half duplex bus with data transfers of 0 to 100 kbit/s in the standard mode and 0 to 400 kbit/s in the Fast mode. In addition to this a High-speed (Hs) mode supports data-transfers up to 3.4 Mbit/s. I²C provides a protocol that allows devices to communicate directly with each other via two wires. One line is responsible for the clock synchronization (SCL), the other is responsible for the data transfer (SDA). The number of devices connected to the I²C-bus is limited only by a maximum bus capacity of 400 pF. Each device is recognized by a unique address.

The two bi-directional bus lines - a serial data line and a serial clock line - carry information between the devices connected to the bus. They are connected to a positive supply voltage via pull-up resistors. The output stages of devices must have an open drain to perform the required wired-AND function. One line is pulled low when one of the open-drain transistors is selected. Otherwise high impedance state is asserted to the line. External pull up resistors are necessary to lift the level to HIGH.

The I²C specification defines a master/slave relationship where each of the devices can work either as transmitter or receiver depending on the function of the device. This functionality is set in the initialization procedure of each module.

Features

- Four identical I²C controller are embedded
- Master mode supported
- Compatible with [9][1)]
- Data transfer in standard- (0-100 kBaud), fast- (0-400 kBaud), fast mode plus (0-1 Mbps) and high-speed mode (0-3.4 Mbps) supported
- APB interface system integration, also including the FIFO concept to allow an unload of the bus system by buffering the data.

Also the module on loads the CPU of low level tasks like:

- (De)Serialization of the bus data
- Generation of start and stop conditions
- Bus access arbitration in multi-master mode
- Bulk transfer mode
- Programmable SDA hold time
- Handles Bit and Byte waiting at all bus speeds

Generic Features of the I²C

These generic features depend on the final implementation of each instance.

- FIFO depth. Configurable for 32 stages are supported.

1) For deviations of timing values compared to the Philips Semiconductors (now NXP Semiconductors) specifications, see **Chapter 15 Electrical Characteristics of URX851/URX850/MxL25641**.

9.3.2 Instances of I²C

Table 118 describes the four instances of I²C in the chip.

Table 118 Instances of I²C

	I²C 0	I²C 1,2,3
Mode	Master	Master
Baud Rate	100 Kbaud 400 Kbaud 1000 Kbaud 3400 Kbaud	100 Kbaud 400 Kbaud 1000 Kbaud 3400 Kbaud
Pin Assignment	SCL: GPIO xx SDA: GPIO xx	SCL: GPIO xx SDA: GPIO xx
Target Applications	All applications: Controlling power regulator chip for CPU power rails; placed in always-on RoC domain	All applications

9.4 GPIO

The module includes four parallel GPIO ports.

9.4.1 Features

The GPIO ports support:

- Number of GPIOs
  - URX851/URX850: Up to 104 general purpose I/Os
  - URX851/URX850: Up to 92 general purpose I/Os
- Each GPIO pin can be interrupt input
- GPIO programming support bit-wise write (no need read-modify-write)
- Input/output selectable
- Input function
  - Tristate
  - High-impedance with a weak pull-up device
  - High-impedance with a weak pull-down device
  - Schmitt-Trigger detection (always enabled)
- Output functions
  - Push-pull (optional pull-up or pull-down)
  - Open drain with a weak internal pull-up
  - Open drain with an external pull-up

9.4.2 GPIO Functionality

Figure 65 shows a general block diagram of a port line. Each port line is equipped with a number of control and data bits, enabling very flexible usage of the line.

Each port pin can be configured for input or output operation.

In input mode (default after reset), the output driver is switched off (high-impedance). The actual voltage level present at the port pin is translated into a logic 0 or 1 via a Schmitt-Trigger device and can be read via the read only register `Px_IN`.

In output mode, the output driver is activated and drives the value supplied through the multiplexer to the port pin. For the GPIO function, switching between input and output mode is accomplished through the `Px_DIR` register, which enables or disables the output driver. When the same pin is used for an alternative function, the output driver may be controlled by the function module.

The output multiplexer in front of the output driver enables the port output function to be used for different purposes. When the pin is used as general purpose output, the multiplexer is switched by software to the output data register `Px_OUT`. The software can set or clear the bit in `Px_OUT`, and it can directly influence the state of the port pin. When the on-chip peripheral units use the pin for output signals, alternate output lines can be switched via the multiplexer to the output driver circuitry.

Latch `Px_IN` is provided for input functions of the on-chip peripheral units. Its input is connected to the output of the input Schmitt-Trigger. Further, an input signal can be connected directly to the various inputs of the peripheral units (AltDataIn). The function of the input line from the pin to the input latch `Px_IN` and to AltDataIn is independent of the port pin operates as input or output. When the port is in output mode, the level of the pin can be read by software via latch `Px_IN` or a peripheral can use the pin level as an input. This offers additional advantages in an application.

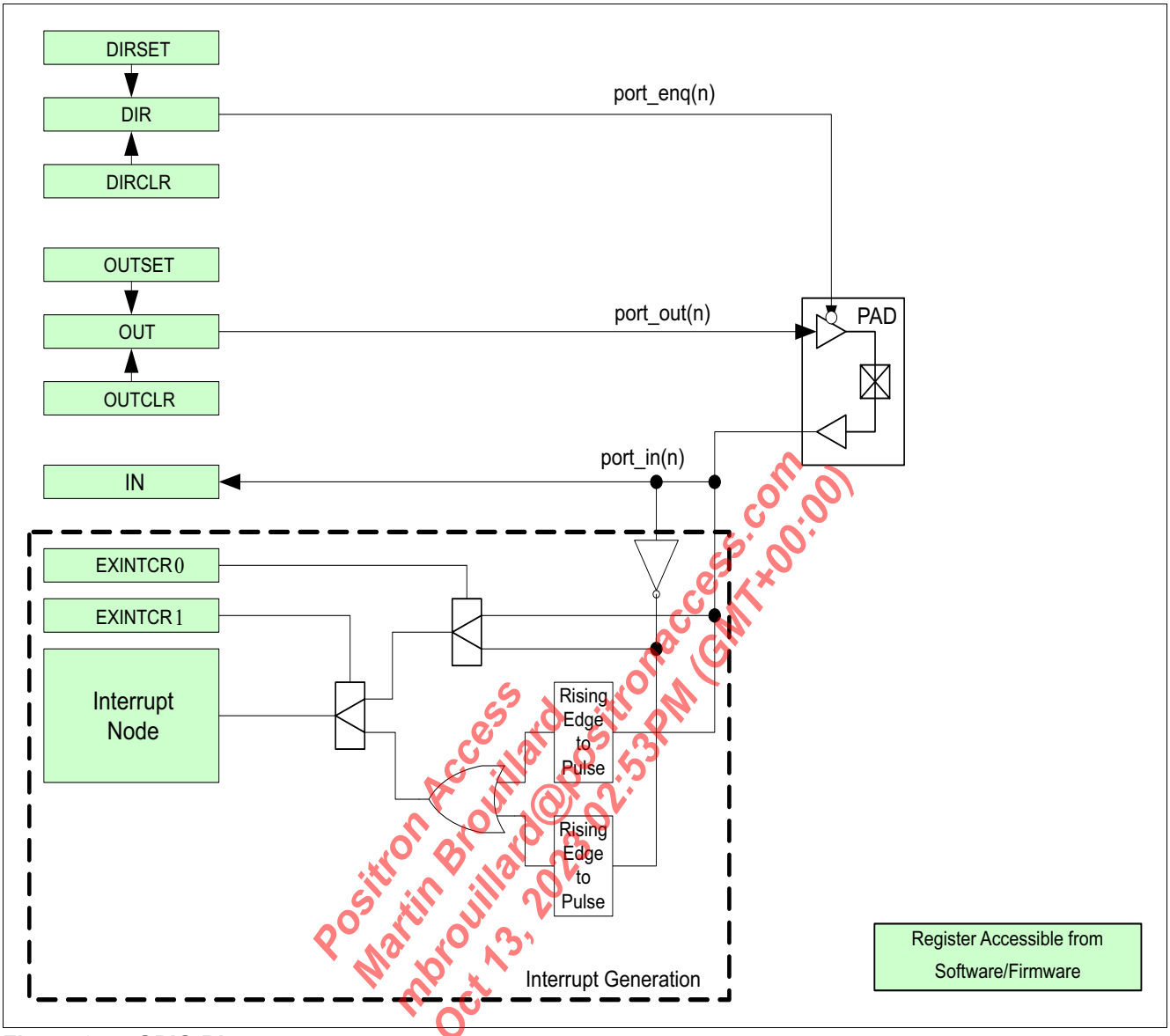


Figure 65 GPIO Bit

9.4.3 Pad Control Overview

This block provides the software/firmware interface for the static pad control for up to 32 pads.

Per Pad Features

- Pull-up enable
- Pull-down enable
- Slew rate control (1-bit)
- Drive current control (2-bit): 2, 4, 8, or 12 mA
- Pad multiplexer control (up to 3-bit, 2 bits used for four possible alternative function)

9.4.4 External Interface: URX851/URX850

Attention: The pin functionality in Table 119 highlighted in bold indicates the pin functionality after reset.

Table 119 Port 0 Functions

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
0	**GPIO used as input**	GPIO	**PORTMUXC_00**:00	**DIR_0**
	GPIO used as output			
	GPC1_1588	CGU	01	
	–		10	
	SPI1_CS2_N	SPI	11	
1	**GPIO used as input**	GPIO	**PORTMUXC_01**:00	**DIR_0**
	GPIO used as output			
	USB_C_PWR_EN1	UVD	01	
	–		10	
	SPI2_CS2_N	SPI	11	
2	**GPIO used as input**	GPIO	**PORTMUXC_02**:00	**DIR_0**
	GPIO used as output			
	USB_C_PWR_EN2	UVD	10	
	–	-	11	
	SPI1_CS4_N	SPI	01	
3	**GPIO used as input**	GPIO	**PORTMUXC_03**:00	**DIR_0**
	GPIO used as output			
	CLKOUT2	CGU	01	
	40 MHz used as output			
	–	-	10	
	I2S_CLK1	I2S	11	
	–	--		
4	GPIO used as input	GPIO	**PORTMUXC_04**:00	**DIR_0**
	GPIO used as output			
	LED_ST	LEDC	01	
	LED_DD0	LEDC	10	
	SPI_Slv_clk	SPI	11	
5	GPIO used as input	GPIO	**PORTMUXC_05**:00	**DIR_0**
	GPIO used as output			
	LED_D	LEDC	01	
	LED_DD1	LEDC	10	
	SPI_Slv_In	SPI	11	

Table 119 Port 0 Functions (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
6	GPIO used as input	GPIO	**PORTMUXC_06**:00	**DIR_0**
	GPIO used as output			
	LED_SH	LEDC	01	
	LED_DD2	LEDC	10	
	SPI_Slv_Out	SPI	11	
7	**GPIO used as input**	GPIO	**PORTMUXC_07**:00	**DIR_0**
	GPIO used as output			
	LED_SH1	LEDC	01	
	CLKOUT1	CGU	10	
	FAN_CTRL_In	PWM	11	
8	**GPIO used as input**	GPIO	**PORTMUXC_08**:00	**DIR_0**
	GPIO used as output			
	C55_SPI1_CS1	C55Voice	01	
	CLKOUT0 (8.192 MHz)	CGU	10	
	-	-	11	
9	GPIO used as input	GPIO	**PORTMUXC_09**:00	**DIR_0**
	GPIO used as output			
	PON_TX_SD	PON	01	
	SPI2_CS1_N	SPI	10	
	SPI_Slv_CS_N	SPI	11	
10	**GPIO used as input**	GPIO	**PORTMUXC_10**:00	**DIR_0**
	GPIO used as output			
	SPI1_TX	SPI1	01	
	SSI1_TX	SSI	10	
	UART2_TX	UART	11	
11	**GPIO used as input**	GPIO	**PORTMUXC_11**:00	**DIR_0**
	GPIO used as output			
	SPI1_RX	SPI1	01	
	SSI1_RX	SSI	10	
	UART2_RX	UART	11	
12	GPIO used as input	GPIO	**PORTMUXC_12**	**DIR_0**
	GPIO used as output			
	I2C_SDA1-PMIC	I2C	10	
	SPI0_CS6_N	SPI	11	

Table 119 Port 0 Functions (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
13	GPIO used as input	GPIO	**PORTMUXC_13**:00	**DIR_0**
	GPIO used as output			
	NAND_ALE	EBU	01	
	–	–	10	
	SPI1_CS3_N	SPI	11	
14	**GPIO used as input**	GPIO	**PORTMUXC_14**:00	**DIR_0**
	GPIO used as output			
	SPI1_CS0_N	SPI1	01	
	SLIC_RST1	SSI	10	
	UART2_RTS	UART	11	
15	**GPIO used as input**	GPIO	**PORTMUXC_15**:00	**DIR_0**
	GPIO used as output			
	SPI0_CS1_N	SPI0	01	
	$\overline{\text{SPI0\_CS1}}$ is used as output			
	–		10	
	C55_SPI1_CS0	C55 voice subsystem	11	
16	**GPIO used as input**	GPIO	**PORTMUXC_16**:00	**DIR_0**
	GPIO used as output			
	SPI0_DIN	SPI0	01	
	–			
	C55_SPI1_RX	C55 voice subsystem	11	
17	**GPIO used as input**	GPIO	**PORTMUXC_17**:00	**DIR_0**
	GPIO used as output			
	SPI0_DOUT	SPI0	01	
	–			
	C55_SPI1_TX	C55 voice subsystem	11	
18	**GPIO used as input**	GPIO	**PORTMUXC_18**:00	**DIR_0**
	GPIO used as output			
	SPI0_CLK	SPI0	01	
	–		10	
	C55_SPI1_CLK	C55 voice subsystem	11	
19	**GPIO used as input**	GPIO	**PORTMUXC_19**:00	**DIR_0**
	GPIO used as output			
	SPI1_CLK	SPI1	01	
	SSI1_CLK	SSI	10	
	UART2_CTS	UART	11	

Table 119 Port 0 Functions (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
20	GPIO Used as Input	GPIO	**PORTMUXC_20**	**DIR_0**
	GPIO used as output			
	–	–	01	
	I2C_SCL1-PMIC	I2C	10	
	SPI1_CS5	SPI3	11	
21	**GPIO used as input**	GPIO	**PORTMUXC_21**:00	**DIR_0**
	GPIO used as output			
	–	--	01	
	I2C_SDA1	I2C	10	
	–	--	11	
22	**GPIO used as input**	GPIO	**PORTMUXC_22**:00	**DIR_0**
	GPIO used as output			
	I2C_SCL1	I2C	01	
	–	--	10	
	–	--	11	
23	GPIO used as input	GPIO	**PORTMUXC_23**:00	**DIR_0**
	GPIO used as output			
	NAND_CS1	NAND	01	
	LED_ST1	LEDC	10	
	–	--	11	
24	GPIO used as input	GPIO	**PORTMUXC_24**:00	**DIR_0**
	GPIO used as output			
	NAND_CLE	EBU	01	
	–	--	10	
	SPI0_CS4	SPI	11	
25	GPIO used as input	GPIO	**PORTMUXC_25**	**DIR_0**
	GPIO used as output			
	NAND_BC0	EBU	01	
	–	--	10	
	GPC2_1588	CGU	11	
26	GPIO used as input	GPIO	**PORTMUXC_26**	**DIR_0**
	GPIO used as output			
	NAND_BC1	EBU	01	
	–	–	10	
	–	–	11	

Table 119 Port 0 Functions (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
27	**GPIO used as input**	GPIO	**PORTMUXC_27**	**DIR_0**
	GPIO used as output			
	SLIC_RST0	SSI	01	
	SPI1_CS1_N	SPI	10	
	C55_RESET0	C55 Voice	11	
28	**GPIO used as input**	GPIO	**PORTMUXC_28**:00	**DIR_0**
	GPIO used as output			
	-	-	01	
	TDM1_FSC	TDM	10	
	–	--	11	
29	**GPIO used as input**	GPIO	00	**DIR_0**
	GPIO used as output			
	-	-	**PORTMUXC_29**:01	
	TDM1_DO	TDM	10	
	–	--	11	
30	**GPIO used as input**	GPIO	**PORTMUXC_30**:00	**DIR_0**
	GPIO used as output			
	-	-	01	
	TDM1_DI	TDM	10	
	–	--	11	
31	**GPIO used as input**	GPIO	**PORTMUXC_31**:00	**DIR_0**
	GPIO used as output			
	-	-	01	
	TDM1_DCL	TDM	10	
	–	--	11	

Table 120 Port 1,2,3,4 Functions

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
32	**GPIO used as input**	GPIO	**PORTMUXC_32**:00	**DIR_1**
	GPIO used as output			
	LED_D1	LEDC	01	
	I2S_WA1	I2S1	10	
	TDM0_FSC	TDM0	11	

Table 120 Port 1,2,3,4 Functions (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
33	**GPIO used as input**	GPIO	**PORTMUXC_33**:00	**DIR_1**
	GPIO used as output			
	–	--	01	
	TDM0_DO	TDM0	10	
	I2S_TX1	I2S1	11	
34	**GPIO used as input**	GPIO	**PORTMUXC_34**:00	**DIR_1**
	GPIO used as output			
	SSI0_TX	SSI	01	
	C55_SPI1_TX	C55 Voice	10	
	-	-	11	
35	**GPIO used as input**	GPIO	**PORTMUXC_35**:00	**DIR_1**
	GPIO used as output			
	SSI0_RX	SSI	01	
	C55_SPI1_RX	C55 Vcodec	10	
	-	-	11	
36	**GPIO used as input**	GPIO	**PORTMUXC_36**:00	**DIR_1**
	GPIO used as output			
	SSI0_CLK	SSI	01	
	C55_SPI1_CLK	C55 Vcodec	10	
	-	-	11	
37	**GPIO used as input**	GPIO	**PORTMUXC_37**	**DIR_1**
	GPIO used as output			
	–		01	
	TDM0_DI	TDM0	10	
	I2S_RX1	I2S1	11	
38	**GPIO used as input**	GPIO	**PORTMUXC_38**	**DIR_1**
	GPIO used as output			
	–		01	
	TDM0_DCL	TDM0	10	
	I2S_CLK1	I2S1	11	
39	**GPIO used as input**	GPIO	**PORTMUXC_39**	**DIR_1**
	GPIO used as output			
	PON_LOS	PON	01	
	SPI3_CS1_N	SPI	10	

Table 120 Port 1,2,3,4 Functions (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
40	**GPIO used as input**	GPIO	**PORTMUXC_40**	**DIR_1**
	GPIO used as output			
	–		01	
	LED_DD3	LEDC	10	
	–		11	
41	**GPIO used as input**	GPIO	**PORTMUXC_41**	**DIR_1**
	MDIO_WANC	MDIO	01	
	FAN_CTRL_In	PWM	10	
	LED_DD4	LEDC	11	
42	**GPIO used as input**	GPIO	**PORTMUXC_42**:00	**DIR_1**
	GPIO used as output			
	MDIO_WAN	MDIO	01	
	LED_DD5	LEDC	10	
	FAN_CTRL_OUT	PWM	11	
43	**GPIO used as input**	GPIO	**PORTMUXC_43**:00	**DIR_1**
	GPIO used as output			
	–	-	01	
	SATA_MP_SWITCH	SATA0	10	
	–		11	
44	**GPIO used as input**	GPIO	**PORTMUXC_44**	**DIR_1**
	GPIO used as output			
	I2S_CLK0	I2S	01	
	SPI2_CLK	SPI	10	
45	**GPIO used as input**	GPIO	**PORTMUXC_45**	**DIR_1**
	GPIO used as output			
	I2S_WA0	I2S	01	
	SPI2_TX	SPI	10	
46	**GPIO used as input**	GPIO	**PORTMUXC_46**	**DIR_1**
	GPIO used as output			
	I2S_RX0	I2S	01	
	SPI2_RX	SPI	10	
47	**GPIO used as input**	GPIO	**PORTMUXC_47**	**DIR_1**
	GPIO used as output			
	I2S_TX0	I2S	01	
	SPI2_CS0_N	SPI	10	

Table 120 Port 1,2,3,4 Functions (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
48	GPIO used as input	GPIO	**PORTMUXC_48**:00	**DIR_1**
	GPIO used as output			
	ND_RD_BY_N	EBU	01	
	–	--	10	
	–	--	11	
49	GPIO used as input	GPIO	**PORTMUXC_49**:00	**DIR_1**
	GPIO used as output			
	ND_RD_N	EBU	01	
	–	--	10	
	–	--	11	
50	GPIO used as input	GPIO	**PORTMUXC_50**:00	**DIR_1**
	GPIO used as output			
	D1	EBU	01	
	–	--	10	
	QSPI_D1	QSPI	11	
51	GPIO used as input	GPIO	**PORTMUXC_51**:00	**DIR_1**
	GPIO used as output			
	D0	EBU	01	
	–	--	10	
	QSPI_D0	QSPI	11	
52	GPIO used as input	GPIO	**PORTMUXC_52**:00	**DIR_1**
	GPIO used as output			
	D2	EBU	01	
	–	--	10	
	QSPI_D2	QSPI	11	
53	GPIO used as input	GPIO	**PORTMUXC_53**:00	**DIR_1**
	GPIO used as output			
	D7	EBU	01	
	–	--	10	
	QSPI_D3	QSPI	11	
54	GPIO used as input	GPIO	**PORTMUXC_54**:00	**DIR_1**
	GPIO used as output			
	D6	EBU	01	
	–	--	10	
	QSPI_CLK	QSPI	11	

Table 120 Port 1,2,3,4 Functions (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
55	GPIO used as input	GPIO	**PORTMUXC_55**:00	**DIR_1**
	GPIO used as output			
	D5	EBU	01	
	–	--	10	
	QSPI_RST0	QSP	11	
56	GPIO used as input	GPIO	**PORTMUXC_56**:00	**DIR_1**
	GPIO used as output			
	D4	EBU	01	
	–	--	10	
	QSPI_CS0	QSP	11	
57	GPIO used as input	GPIO	**PORTMUXC_57**:00	**DIR_1**
	GPIO used as output			
	D3	EBU	01	
	–	--	10	
	QSPI_CS1	QSP	11	
58	GPIO used as input	GPIO	**PORTMUXC_58**	**DIR_1**
	GPIO used as output			
	NAND_CS0	EBU	01	
	–	--	10	
	QSPI_RST1	QSP	11	
59	GPIO used as input	GPIO	**PORTMUXC_59**:00	**DIR_1**
	GPIO used as output			
	NAND_WR_N	EBU	01	
	–	--	10	
	–	--	11	
60	GPIO used as input	GPIO	**PORTMUXC_60**:00	**DIR_1**
	GPIO used as output			
	NAND_WP_N	EBU	01	
	SLIC2C55_INT1	C55	10	
	I2S_TX1	I2S	11	
61	GPIO used as input	GPIO	**PORTMUXC_61**:00	**DIR_1**
	GPIO used as output			
	NAND_SE	EBU	01	
	SLIC2C55_INT0	C55	10	
	I2S_WA1	I2S	11	

Table 120 Port 1,2,3,4 Functions (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
62	**GPIO used as input**	GPIO	**PORTMUXC_62**	**DIR_1**
	GPIO used as output			
	I2C3_SCL	I2C	01	
	–		10	
	I2S_RX1	I2S	11	
63	**GPIO used as input**	GPIO	**PORTMUXC_63**	**DIR_1**
	GPIO used as output			
	I2C3_SDA	I2C	01	
	–		10	
	–		11	
64	GPIO used as input		**PORTMUXC_64**	**DIR_1**
	GPIO used as output			
	UART_RX0	UART	01	
	–		10	
	–	--	11	
65	GPIO used as input	GPIO	**PORTMUXC_65**	**DIR_1**
	GPIO used as output			
	UART_TX0	UART	01	
	–		10	
	–	--	11	
66	**GPIO used as input**	GPIO	**PORTMUXC_66**	**DIR_1**
	GPIO used as output			
	UART1_RX	UART	01	
	C55_TDM2_FSC	C55	10	
	TDM2_FSC	TDM2	11	
67	**GPIO used as input**	GPIO	**PORTMUXC_67**	**DIR_1**
	GPIO used as output			
	UART1_TX	UART	01	
	C55_TDM2_DO	C55	10	
	TDM2_DO	TDM2	11	
68	**GPIO used as input**	GPIO	**PORTMUXC_68**	**DIR_1**
	GPIO used as output			
	UART2_RX	UART	01	
	C55_TDM2_DI	C55	10	
	TDM2_DI	TDM2	11	

Table 120 Port 1,2,3,4 Functions (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
69	**GPIO used as input**	GPIO	**PORTMUXC_69**	**DIR_1**
	GPIO used as output			
	UART2_TX	UART	01	
	C55_TDM2_DCL	C55	10	
	TDM2_DCL	TDM2	11	
70	GPIO used as input	GPIO	**PORTMUXC_70**	**DIR_1**
	GPIO used as output			
	I2C_SDA2	I2C	01	
	UART3_RX		10	
	SPI3_CS0_N	SPI	11	
71	GPIO used as input	GPIO	**PORTMUXC_71**	**DIR_1**
	GPIO used as output			
	I2C_SCL2	I2C	01	
	UART3_TX	UART	10	
	SPI3_TX	SPI	11	
72	GPIO used as input	GPIO	**PORTMUXC_72**	**DIR_1**
	GPIO used as output			
	TX-FAULT	PON	01	
	UART3_RTS	UART	10	
	SPI3_RX		11	
73	GPIO used as input	GPIO	**PORTMUXC_73**	**DIR_1**
	GPIO used as output			
	TX-Disable	PON	01	
	UART3_CTS	UART	10	
	SPI3_CLK	SPI	11	
74	GPIO used as input	GPIO	**PORTMUXC_74**	**DIR_1**
	GPIO used as output			
	NAND_D13	EBU	01	
	SATA0_PHY_DEVSLP	SATA0	10	
	SDIO_CLK		11	
75	GPIO used as input	GPIO	**PORTMUXC_75**	**DIR_1**
	GPIO used as output			
	NAND_D12	EBU	01	
	SATA1_PHY_DEVSLP	SATA1	10	
	SDIO_CMD	SDIO	11	

Table 120 Port 1,2,3,4 Functions (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
76	GPIO used as input	GPIO	PORTMUXC_76	DIR_1
	GPIO used as output			
	NAND_D10	EBU	01	
	SATA2_MP_SWH	SATA2	10	
	SDIO_D2	SDIO	11	
77	GPIO used as input	GPIO	PORTMUXC_77	DIR_1
	GPIO used as output			
	NAND_D11	EBU	01	
	SATA2_CP_Detect	SATA2	10	
	SDIO_D3	SDIO	11	
78	GPIO used as input	GPIO	PORTMUXC_78	DIR_1
	GPIO used as output			
	NAND_D9	EBU	01	
	SATA3_MP_SWH	SATA3	10	
	SDIO_D1	SDIO	11	
79	GPIO used as input	GPIO	PORTMUXC_79	DIR_1
	GPIO used as output			
	NAND_D8	EBU	01	
	SATA3_CP_Detect	SATA3	10	
	SDIO_D0	SDIO	11	
80	GPIO used as input	GPIO		DIR_1
	GPIO used as output			
	NAND_D14	EBU	01	
	SATA1_MP_SWH	SATA1	10	
	SDIO_CD	SDIO	11	
81	GPIO used as input	GPIO		DIR_1
	GPIO used as output			
	NAND_D15	EBU	01	
	SATA1_CP_Detect	SATA1	10	
	SDIO_WP	SDIO	11	
82	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	–		01	
	SATA0_CP_Detect	SATA0	10	
	Bright_Dect		11	

Table 120 Port 1,2,3,4 Functions (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
83	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	–		01	
	PCI10_CLK_REQ		10	
	–		11	
84	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	–		01	
	PCI11_CLK_REQ		10	
	–		11	
85	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	–		01	
	PCI20_CLK_REQ		10	
	–		11	
86	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	–		01	
	PCI21_CLK_REQ		10	
	–		11	
87	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	–		01	
	PCI30_CLK_REQ		10	
	–		11	
88	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	–		01	
	PCI31_CLK_REQ		10	
	–		11	
89	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	–		01	
	PCI40_CLK_REQ		10	
	–		11	

Table 120 Port 1,2,3,4 Functions (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
90	GPIO used as input	GPIO		DIR_1
	GPIO used as output			
	PMIC-GPIO5	EPU	01	
	PCI41_CLK_REQ		10	
	–		11	
91	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	PPS1_IN_OUT	CGU	01	
	LED_DD6	LEDC	10	
	–		11	
92	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	PPS2_IN_OUT	CGU	01	
	LED_DD7	LEDC	10	
	–		11	
93	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	CLK32K		01	
	NTR	CGU	10	
	LED_DD9	LEDC	11	
94	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	MDIO1_CLK	GSWIP	01	
	–		10	
	–		11	
95	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	MDIO_LAN	GSWIP	01	
	–		10	
	–		11	
96	GPIO used as input	GPIO		DIR_1
	GPIO used as output			
	PMIC_GPIO0	EPU	01	
	–		10	
	–		11	

Table 120 Port 1,2,3,4 Functions (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
97	GPIO used as input	GPIO		DIR_1
	GPIO used as output			
	PMIC_GPIO1	EPU	01	
	–		10	
	–		11	
98	GPIO used as input	GPIO		DIR_1
	GPIO used as output			
	PMIC_GPIO2	EPU	01	
	–		10	
	–		11	
99	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	–		01	
	LED_DD8	LEDC	10	
	SATA2_PHY_DEVSLP	SATA3	11	
100	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	C55_SCC1_RESET1	C55 SCC1	01	
	–		10	
	–		11	
101	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	C55_SPI0_CS1	C55 SPI0	01	
			10	
	SATA3_PHY_DEVSLP		11	
102	GPIO used as input	GPIO		DIR_1
	GPIO used as output			
	PMIC_GPIO3	EPU	01	
	–		10	
	–		11	
103	GPIO used as input	GPIO		DIR_1
	GPIO used as output			
	PMIC_GPIO4	EPU	01	
	–		10	
	–		11	

9.4.4.1 Top Level Multiplexing Addition to GPIO Mux

These pins have extra multiplexing functions in addition to GPIO multiplexing. They are controlled by chip_top register bits.

Table 121 Chip Top Multiplexing

Pin	Top Level Function 1	Top Level Function 2
GPIO0	C55_SPI1_CS2	–
GPIO2	C55_SPI1_CS4	–
GPIO10	C55_SPI0_TX	C55_UART2_TX
GPIO11	C55_SPI0_RX	C55_UART2_RX
GPIO13	C55_SPI0_CS3	–
GPIO14	C55_SPI0_CS0	C55_UART2_RTS
GPIO19	C55_SPI0_CLK	C55_UART2_CTS
GPIO20	C55_SPI0_CS5	–
GPIO28	C55_TDM1_FSC	C55_TDM0_FSC
GPIO29	C55_TDM1_DO	C55_TDM0_DO
GPIO30	C55_TDM1_DI	C55_TDM0_DI
GPIO31	C55_TDM1_DCL	C55_TDM0_DCL
GPIO32	C55_TDM0_FSC	–
GPIO33	C55_TDM0_DO	–
GPIO37	C55_TDM0_DI	–
GPIO38	C55_TDM0_DCL	–
GPIO41	TEP_GPIO1	–
GPIO42	TEP_GPIO2	–
GPIO43	TEP_GPIO3	–
GPIO44	TEP_GPIO4	UART3_RTS
GPIO45	TEP_GPIO5	UART3_TX
GPIO46	TEP_GPIO6	UART3_RX
GPIO47	TEP_GPIO7	UART3_CTS
GPIO66	NTR (double mapped CLK32K)	–
GPIO67	GPC1_1588 (double mapped GPIO0)	–
GPIO68	PPS1 (double mapped GPIO91)	–
GPIO69	PON_RX_LOS	–
GPIO70	PON_I2C_SDA	–
GPIO71	PON_I2C_SCL	–
GPIO75	TRC4	–
GPIO76	TRC2	–
GPIO77	TRC3	–
GPIO78	TRC1	–
GPIO79	TRC0	–
GPIO80	TRC5	–

Table 121 Chip Top Multiplexing (cont'd)

Pin	Top Level Function 1	Top Level Function 2
GPIO81	TRC6	–
GPIO82	TRC7	–
GPIO83	TRC8	–
GPIO84	TRC9	–
GPIO85	TRC10	–
GPIO86	TRC11	–
GPIO87	TRC12	–
GPIO88	TRC13	–
GPIO89	TRC14	–
GPIO90	TRC15	–
GPIO91	TRC_Data_Valid	–

9.4.5 External Interface: MxL25641

Attention: The pin functionality in Table 119 highlighted in bold indicates the pin functionality after reset.

Table 122 Port 0 Functions MxL25641

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
0	**GPIO used as input**	GPIO	**PORTMUXC_00**:00	**DIR_0**
	GPIO used as output			
	GPC1_1588	CGU	01	
	–		10	
	SPI1_CS2_N	SPI	11	
1	**GPIO used as input**	GPIO	**PORTMUXC_01**:00	**DIR_0**
	GPIO used as output			
	--	--	01	
	–		10	
	SPI2_CS2_N	SPI	11	
2	**GPIO used as input**	GPIO	**PORTMUXC_02**:00	**DIR_0**
	GPIO used as output			
	--	--	10	
	–	-	11	
	SPI1_CS4_N	SPI	01	
3	**GPIO used as input**	GPIO	**PORTMUXC_03**:00	**DIR_0**
	GPIO used as output			
	CLKOUT2	CGU	01	
	40 MHz used as output			
	–	-	10	
	I2S_CLK1	I2S	11	
	–	--		
4	GPIO used as input	GPIO	**PORTMUXC_04**:00	**DIR_0**
	GPIO used as output			
	LED_ST	LEDC	01	
	LED_DD0	LEDC	10	
	SPI_Slv_clk	SPI	11	
5	GPIO used as input	GPIO	**PORTMUXC_05**:00	**DIR_0**
	GPIO used as output			
	LED_D	LEDC	01	
	LED_DD1	LEDC	10	
	SPI_Slv_In	SPI	11	

Table 122 Port 0 Functions MxL25641 (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
6	GPIO used as input	GPIO	**PORTMUXC_06**:00	**DIR_0**
	GPIO used as output			
	LED_SH	LEDC	01	
	LED_DD2	LEDC	10	
	SPI_Slv_Out	SPI	11	
7	**GPIO used as input**	GPIO	**PORTMUXC_07**:00	**DIR_0**
	GPIO used as output			
	LED_SH1	LEDC	01	
	CLKOUT1	CGU	10	
	FAN_CTRL_In	PWM	11	
8	**GPIO used as input**	GPIO	**PORTMUXC_08**:00	**DIR_0**
	GPIO used as output			
	C55_SPI1_CS1	C55Voice	01	
	CLKOUT0 (8.192 MHz)	CGU	10	
	-	-	11	
9	GPIO used as input	GPIO	**PORTMUXC_09**:00	**DIR_0**
	GPIO used as output			
	PON_TX_SD	PON	01	
	SPI2_CS1_N	SPI	10	
	SPI_Slv_CS_N	SPI	11	
10	**GPIO used as input**	GPIO	**PORTMUXC_10**:00	**DIR_0**
	GPIO used as output			
	SPI1_TX	SPI1	01	
	SSI1_TX	SSI	10	
	UART2_TX	UART	11	
11	**GPIO used as input**	GPIO	**PORTMUXC_11**:00	**DIR_0**
	GPIO used as output			
	SPI1_RX	SPI1	01	
	SSI1_RX	SSI	10	
	UART2_RX	UART	11	
12	GPIO used as input	GPIO	**PORTMUXC_12**	**DIR_0**
	GPIO used as output			
	I2C_SDA1-PMIC	I2C	10	
	SPI0_CS6_N	SPI	11	

Table 122 Port 0 Functions MxL25641 (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
13	GPIO used as input	GPIO	**PORTMUXC_13**:00	**DIR_0**
	GPIO used as output			
	NAND_ALE	EBU	01	
	–	–	10	
	SPI1_CS3_N	SPI	11	
14	**GPIO used as input**	GPIO	**PORTMUXC_14**:00	**DIR_0**
	GPIO used as output			
	SPI1_CS0_N	SPI1	01	
	SLIC_RST1	SSI	10	
	UART2_RTS	UART	11	
15	**GPIO used as input**	GPIO	**PORTMUXC_15**:00	**DIR_0**
	GPIO used as output			
	SPI0_CS1_N	SPI0	01	
	$\overline{\text{SPI0\_CS1}}$ is used as output			
	–		10	
	C55_SPI1_CS0	C55 voice subsystem	11	
16	**GPIO used as input**	GPIO	**PORTMUXC_16**:00	**DIR_0**
	GPIO used as output			
	SPI0_DIN	SPI0	01	
	–			
	C55_SPI1_RX	C55 voice subsystem	11	
17	**GPIO used as input**	GPIO	**PORTMUXC_17**:00	**DIR_0**
	GPIO used as output			
	SPI0_DOUT	SPI0	01	
	–			
	C55_SPI1_TX	C55 voice subsystem	11	
18	**GPIO used as input**	GPIO	**PORTMUXC_18**:00	**DIR_0**
	GPIO used as output			
	SPI0_CLK	SPI0	01	
	–		10	
	C55_SPI1_CLK	C55 voice subsystem	11	
19	**GPIO used as input**	GPIO	**PORTMUXC_19**:00	**DIR_0**
	GPIO used as output			
	SPI1_CLK	SPI1	01	
	SSI1_CLK	SSI	10	
	UART2_CTS	UART	11	

Table 122 Port 0 Functions MxL25641 (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
20	GPIO Used as Input	GPIO	**PORTMUXC_20**	**DIR_0**
	GPIO used as output			
	–	–	01	
	I2C_SCL1-PMIC	I2C	10	
	SPI1_CS5	SPI3	11	
21	**GPIO used as input**	GPIO	**PORTMUXC_21**:00	**DIR_0**
	GPIO used as output			
	–	--	01	
	I2C_SDA1	I2C	10	
	–	--	11	
22	**GPIO used as input**	GPIO	**PORTMUXC_22**:00	**DIR_0**
	GPIO used as output			
	I2C_SCL1	I2C	01	
	–	--	10	
	–	--	11	
23	GPIO used as input	GPIO	**PORTMUXC_23**:00	**DIR_0**
	GPIO used as output			
	NAND_CS1	NAND	01	
	LED_ST1	LEDC	10	
	–	--	11	
24	GPIO used as input	GPIO	**PORTMUXC_24**:00	**DIR_0**
	GPIO used as output			
	NAND_CLE	EBU	01	
	–	--	10	
	SPI0_CS4	SPI	11	
25	GPIO used as input	GPIO	**PORTMUXC_25**	**DIR_0**
	GPIO used as output			
	NAND_BC0[1)]	EBU	01	
	–	--	10	
	GPC2_1588	CGU	11	
27	**GPIO used as input**	GPIO	**PORTMUXC_27**	**DIR_0**
	GPIO used as output			
	SLIC_RST0	SSI	01	
	SPI1_CS1_N	SPI	10	
	C55_RESET0	C55 Voice	11	

Table 122 Port 0 Functions MxL25641 (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
28	**GPIO used as input**	GPIO	**PORTMUXC_28**:00	**DIR_0**
	GPIO used as output			
	-	-	01	
	TDM1_FSC	TDM	10	
	–	--	11	
29	**GPIO used as input**	GPIO	00	**DIR_0**
	GPIO used as output			
	-	-	**PORTMUXC_29**:01	
	TDM1_DO	TDM	10	
	–	--	11	
30	**GPIO used as input**	GPIO	**PORTMUXC_30**:00	**DIR_0**
	GPIO used as output			
	-	-	01	
	TDM1_DI	TDM	10	
	–	--	11	
31	**GPIO used as input**	GPIO	**PORTMUXC_31**:00	**DIR_0**
	GPIO used as output			
	-	-	01	
	TDM1_DCL	TDM	10	
	–	--	11	

Table 123 Port 1,2,3,4 Functions MxL25641

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
32	**GPIO used as input**	GPIO	**PORTMUXC_32**:00	**DIR_1**
	GPIO used as output			
	LED_D1	LEDC	01	
	I2S_WA1	I2S1	10	
	TDM0_FSC	TDM0	11	
33	**GPIO used as input**	GPIO	**PORTMUXC_33**:00	**DIR_1**
	GPIO used as output			
	–	--	01	
	TDM0_DO	TDM0	10	
	I2S_TX1	I2S1	11	
34	**GPIO used as input**	GPIO	**PORTMUXC_34**:00	**DIR_1**
	GPIO used as output			
	SSI0_TX	SSI	01	
	C55_SPI1_TX	C55 Voice	10	
	-	-	11	
35	**GPIO used as input**	GPIO	**PORTMUXC_35**:00	**DIR_1**
	GPIO used as output			
	SSI0_RX	SSI	01	
	C55_SPI1_RX	C55 Vcodec	10	
	-	-	11	
36	**GPIO used as input**	GPIO	**PORTMUXC_36**:00	**DIR_1**
	GPIO used as output			
	SSI0_CLK	SSI	01	
	C55_SPI1_CLK	C55 Vcodec	10	
	-	-	11	
37	**GPIO used as input**	GPIO	**PORTMUXC_37**	**DIR_1**
	GPIO used as output			
	–		01	
	TDM0_DI	TDM0	10	
	I2S_RX1	I2S1	11	
38	**GPIO used as input**	GPIO	**PORTMUXC_38**	**DIR_1**
	GPIO used as output			
	–		01	
	TDM0_DCL	TDM0	10	
	I2S_CLK1	I2S1	11	

Table 123 Port 1,2,3,4 Functions MxL25641 (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
39	**GPIO used as input**	GPIO	**PORTMUXC_39**	**DIR_1**
	GPIO used as output			
	PON_LOS	PON	01	
	SPI3_CS1_N	SPI	10	
41	**GPIO used as input**	GPIO	**PORTMUXC_41**	**DIR_1**
	MDIO_WANC	MDIO	01	
	FAN_CTRL_In	PWM	10	
	LED_DD4	LEDC	11	
42	**GPIO used as input**	GPIO	**PORTMUXC_42**:00	**DIR_1**
	GPIO used as output			
	MDIO_WAN	MDIO	01	
	LED_DD5	LEDC	10	
	FAN_CTRL_OUT	PWM	11	
44	**GPIO used as input**	GPIO	**PORTMUXC_44**	**DIR_1**
	GPIO used as output			
	I2S_CLK0	I2S	01	
	SPI2_CLK	SPI	10	
45	**GPIO used as input**	GPIO	**PORTMUXC_45**	**DIR_1**
	GPIO used as output			
	I2S_WA0	I2S	01	
	SPI2_TX	SPI	10	
46	**GPIO used as input**	GPIO	**PORTMUXC_46**	**DIR_1**
	GPIO used as output			
	I2S_RX0	I2S	01	
	SPI2_RX	SPI	10	
47	**GPIO used as input**	GPIO	**PORTMUXC_47**	**DIR_1**
	GPIO used as output			
	I2S_TX0	I2S	01	
	SPI2_CS0_N	SPI	10	
48	GPIO used as input	GPIO	**PORTMUXC_48**:00	**DIR_1**
	GPIO used as output			
	ND_RD_BY_N	EBU	01	
	–	--	10	
	–	--	11	

Table 123 Port 1,2,3,4 Functions MxL25641 (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
49	GPIO used as input	GPIO	**PORTMUXC_49**:00	**DIR_1**
	GPIO used as output			
	ND_RD_N	EBU	01	
	–	--	10	
	–	--	11	
50	GPIO used as input	GPIO	**PORTMUXC_50**:00	**DIR_1**
	GPIO used as output			
	D1	EBU	01	
	–	--	10	
	QSPI_D1	QSPI	11	
51	GPIO used as input	GPIO	**PORTMUXC_51**:00	**DIR_1**
	GPIO used as output			
	D0	EBU	01	
	–	--	10	
	QSPI_D0	QSPI	11	
52	GPIO used as input	GPIO	**PORTMUXC_52**:00	**DIR_1**
	GPIO used as output			
	D2	EBU	01	
	–	--	10	
	QSPI_D2	QSPI	11	
53	GPIO used as input	GPIO	**PORTMUXC_53**:00	**DIR_1**
	GPIO used as output			
	D7	EBU	01	
	–	--	10	
	QSPI_D3	QSPI	11	
54	GPIO used as input	GPIO	**PORTMUXC_54**:00	**DIR_1**
	GPIO used as output			
	D6	EBU	01	
	–	--	10	
	QSPI_CLK	QSPI	11	
55	GPIO used as input	GPIO	**PORTMUXC_55**:00	**DIR_1**
	GPIO used as output			
	D5	EBU	01	
	–	--	10	
	QSPI_RST0	QSP	11	

Table 123 Port 1,2,3,4 Functions MxL25641 (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
56	GPIO used as input	GPIO	**PORTMUXC_56**:00	**DIR_1**
	GPIO used as output			
	D4	EBU	01	
	–	--	10	
	QSPI_CS0	QSP	11	
57	GPIO used as input	GPIO	**PORTMUXC_57**:00	**DIR_1**
	GPIO used as output			
	D3	EBU	01	
	–	--	10	
	QSPI_CS1	QSP	11	
58	GPIO used as input	GPIO	**PORTMUXC_58**	**DIR_1**
	GPIO used as output			
	NAND_CS0	EBU	01	
	–	--	10	
	QSPI_RST1	QSP	11	
59	GPIO used as input	GPIO	**PORTMUXC_59**:00	**DIR_1**
	GPIO used as output			
	NAND_WR_N	EBU	01	
	–	--	10	
	–	--	11	
60	GPIO used as input	GPIO	**PORTMUXC_60**:00	**DIR_1**
	GPIO used as output			
	NAND_WP_N	EBU	01	
	SLIC2C55_INT1	C55	10	
	I2S_TX1	I2S	11	
61	GPIO used as input	GPIO	**PORTMUXC_61**:00	**DIR_1**
	GPIO used as output			
	NAND_SE	EBU	01	
	SLIC2C55_INT0	C55	10	
	I2S_WA1	I2S	11	
62	**GPIO used as input**	GPIO	**PORTMUXC_62**	**DIR_1**
	GPIO used as output			
	I2C3_SCL	I2C	01	
	–		10	
	I2S_RX1	I2S	11	

Table 123 Port 1,2,3,4 Functions MxL25641 (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
63	**GPIO used as input**	GPIO	**PORTMUXC_63**	**DIR_1**
	GPIO used as output			
	I2C3_SDA	I2C	01	
	–		10	
	–		11	
64	GPIO used as input		**PORTMUXC_64**	**DIR_1**
	GPIO used as output			
	UART_RX0	UART	01	
	–		10	
	–	--	11	
65	GPIO used as input	GPIO	**PORTMUXC_65**	**DIR_1**
	GPIO used as output			
	UART_TX0	UART	01	
	–		10	
	–	--	11	
66	**GPIO used as input**	GPIO	**PORTMUXC_66**	**DIR_1**
	GPIO used as output			
	UART1_RX	UART	01	
	C55_TDM2_FSC	C55	10	
	TDM2_FSC	TDM2	11	
67	**GPIO used as input**	GPIO	**PORTMUXC_67**	**DIR_1**
	GPIO used as output			
	UART1_TX	UART	01	
	C55_TDM2_DO	C55	10	
	TDM2_DO	TDM2	11	
68	**GPIO used as input**	GPIO	**PORTMUXC_68**	**DIR_1**
	GPIO used as output			
	UART2_RX	UART	01	
	C55_TDM2_DI	C55	10	
	TDM2_DI	TDM2	11	
69	**GPIO used as input**	GPIO	**PORTMUXC_69**	**DIR_1**
	GPIO used as output			
	UART2_TX	UART	01	
	C55_TDM2_DCL	C55	10	
	TDM2_DCL	TDM2	11	

Table 123 Port 1,2,3,4 Functions MxL25641 (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
70	GPIO used as input	GPIO	**PORTMUXC_70**	**DIR_1**
	GPIO used as output			
	I2C_SDA2	I2C	01	
	UART3_RX		10	
	SPI3_CS0_N	SPI	11	
71	GPIO used as input	GPIO	**PORTMUXC_71**	**DIR_1**
	GPIO used as output			
	I2C_SCL2	I2C	01	
	UART3_TX	UART	10	
	SPI3_TX	SPI	11	
72	GPIO used as input	GPIO	**PORTMUXC_72**	**DIR_1**
	GPIO used as output			
	TX-FAULT	PON	01	
	UART3_RTS	UART	10	
	SPI3_RX		11	
73	GPIO used as input	GPIO	**PORTMUXC_73**	**DIR_1**
	GPIO used as output			
	TX-Disable	PON	01	
	UART3_CTS	UART	10	
	SPI3_CLK	SPI	11	
74	GPIO used as input	GPIO	**PORTMUXC_74**	**DIR_1**
	GPIO used as output			
	Test	--	01	
	--	--	10	
	--	--	11	
75	GPIO used as input	GPIO	**PORTMUXC_75**	**DIR_1**
	GPIO used as output			
	Test	--	01	
	--	--	10	
	--	--	11	
77	GPIO used as input	GPIO	**PORTMUXC_77**	**DIR_1**
	GPIO used as output			
	Test	--	01	
	--	--	10	
	--	--	11	

Table 123 Port 1,2,3,4 Functions MxL25641 (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
82	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	–		01	
	--	--	10	
	Bright_Dect		11	
83	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	–		01	
	PCI10_CLK_REQ		10	
	–		11	
84	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	–		01	
	PCI11_CLK_REQ[1)]		10	
	–		11	
85	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	–		01	
	PCI20_CLK_REQ[1)]		10	
	–		11	
86	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	–		01	
	PCI21_CLK_REQ[1)]		10	
	–		11	
90	GPIO used as input	GPIO		DIR_1
	GPIO used as output			
	PMIC-GPIO5	EPU	01	
	--		10	
	–		11	
91	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	PPS1_IN_OUT	CGU	01	
	LED_DD6	LEDC	10	
	–		11	

Table 123 Port 1,2,3,4 Functions MxL25641 (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
92	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	PPS2_IN_OUT	CGU	01	
	LED_DD7	LEDC	10	
	–		11	
93	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	CLK32K		01	
	NTR	CGU	10	
	LED_DD9	LEDC	11	
94	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	MDIO1_CLK	GSWIP	01	
	–		10	
	–		11	
95	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	MDIO_LAN	GSWIP	01	
	–		10	
	–		11	
96	GPIO used as input	GPIO		DIR_1
	GPIO used as output			
	PMIC_GPIO0	EPU	01	
	–		10	
	–		11	
97	GPIO used as input	GPIO		DIR_1
	GPIO used as output			
	PMIC_GPIO1	EPU	01	
	–		10	
	–		11	
98	GPIO used as input	GPIO		DIR_1
	GPIO used as output			
	PMIC_GPIO2	EPU	01	
	–		10	
	–		11	

Table 123 Port 1,2,3,4 Functions MxL25641 (cont'd)

GPIO Pin	Pin Functionality	Associated Register/Module	Alternate Function	Direction Control
100	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	C55_SCC1_RESET1	C55 SCC1	01	
	–		10	
	–		11	
101	**GPIO used as input**	GPIO		DIR_1
	GPIO used as output			
	C55_SPI0_CS1	C55 SPI0	01	
			10	
	--		11	
102	GPIO used as input	GPIO		DIR_1
	GPIO used as output			
	PMIC_GPIO3	EPU	01	
	–		10	
	–		11	
103	GPIO used as input	GPIO		DIR_1
	GPIO used as output			
	PMIC_GPIO4	EPU	01	
	–		10	
	–		11	

9.4.5.1 Top Level Multiplexing Addition to GPIO Mux

These pins have extra multiplexing functions in addition to GPIO multiplexing. They are controlled by `chip_top` register bits.

Table 124 Chip Top Multiplexing

Pin	Top Level Function 1	Top Level Function 2
GPIO0	C55_SPI1_CS2	–
GPIO2	C55_SPI1_CS4	–
GPIO10	C55_SPI0_TX	C55_UART2_TX
GPIO11	C55_SPI0_RX	C55_UART2_RX
GPIO13	C55_SPI0_CS3	–
GPIO14	C55_SPI0_CS0	C55_UART2_RTS
GPIO19	C55_SPI0_CLK	C55_UART2_CTS
GPIO20	C55_SPI0_CS5	–
GPIO28	C55_TDM1_FSC	C55_TDM0_FSC
GPIO29	C55_TDM1_DO	C55_TDM0_DO
GPIO30	C55_TDM1_DI	C55_TDM0_DI
GPIO31	C55_TDM1_DCL	C55_TDM0_DCL
GPIO32	C55_TDM0_FSC	–
GPIO33	C55_TDM0_DO	–
GPIO37	C55_TDM0_DI	–
GPIO38	C55_TDM0_DCL	–
GPIO41	TEP_GPIO1	–
GPIO42	TEP_GPIO2	–
GPIO44	TEP_GPIO4	UART3_RTS
GPIO45	TEP_GPIO5	UART3_TX
GPIO46	TEP_GPIO6	UART3_RX
GPIO47	TEP_GPIO7	UART3_CTS
GPIO66	NTR (double mapped CLK32K)	–
GPIO67	GPC1_1588 (double mapped GPIO0)	–
GPIO68	PPS1 (double mapped GPIO91)	–
GPIO69	PON_RX_LOS	–
GPIO70	PON_I2C_SDA	–
GPIO71	PON_I2C_SCL	–

9.5 Serial Shift Output Controller

Parallel to serial conversion, which is also called Serial Shift Output (SSO) controller, can drive external shift registers for LED outputs, reset or general purpose outputs.

9.5.1 Features

The supported features are:

- SSO Module
  - Module 1 for URX851/URX850/MxL25641 GPO extension
- Common Features:
  - Dedicated blink rate control for Data [31:24], common blink rate for Data[23:0]
  - Individual duty cycle control for Data [31:0]
  - Automatic LED Dimming based on off-chip brightness detector
  - Supports static data pin via CPU or hardware source
  - Supports blinking LED with configurable frequency (linked with GPTC0 T2A)
- GPIO Alternate Function Pins
  - URX851/URX850: A maximum of 10 pins (LED_DD[9:0] which have signals from common feature logic connected to the GPIO alternate function. In this configuration, no shift register is available.
  - MxL25641: A maximum of 8 pins (LED_DD[9],[7:4],[2:0] which have signals from common feature logic connected to the GPIO alternate function. In this configuration, no shift register is available.
  - A maximum of 7 pins via GPIO alternate path and other 25 signals can in parallel go to external shift register. 3 pins have double function and can only be used either for GPIO alternate mode or as shift register pins.
  - Common features are available to be programmed to affect the behavior of the ten mapped pins.
- SSO Controller
  - Interfaces to standard 74 serial shift register, 74X164 or 74X595
  - Allows control of up to 32 data outputs (for example LED) with 3 external pins
  - Shift register update decoupled from blink rate
  - Common features are available to be programmed to affect the behavior of the 32 data outputs shifted out.

9.5.2 Serial Shift Out Controller Functional Description

This section describes the SSO controller functionality.

Note: The SSO controller starts from bit position 0.

Table 125 SSO Module 1 Mapping of HW LED Signal or Internal Signal to SSO_CON3 Register

HW Source	SSO_CON3 bits
OPT_RX_LOS	SSO_CON3[1]
HW_LED[02](SATA0_ACT)	SSO_CON3[2]
HW_LED[13](SATA1_ACT)	SSO_CON3[3]
HW_LED[2](SD_ACT)	SSO_CON3[4]
SSI0_RST	SSO_CON3[5]
HW_LED_1[02](SATA2_ACT)	SSO_CON3[6]
HW_LED_1[13](SATA3_ACT)	SSO_CON3[7]
OPT_TX_PUP	SSO_CON3[13]
OPT_TX_OFF	SSO_CON3[14]
SSI1_RST	SSO_CON3[15]
OPT_GPIO0	SSO_CON3[16]
OPT_GPIO1	SSO_CON3[17]
OPT_GPIO2	SSO_CON3[18]
OPT_GPIO3	SSO_CON3[19]
OPT_GPIO4	SSO_CON3[20]
OPT_GPIO5	SSO_CON3[21]
OPT_GPIO6	SSO_CON3[22]
OPT_GPIO7	SSO_CON3[23]

Table 126 LED to GPIO Connection Register

Parallel LED Output	GPIO Alternative Input
OPT_RX_LOS or CPU Source [1]	LED1
HW_LED[2][02][13] or CPU Source [4:2]	LED4, LED3[1)], LED2
HW_LED_1[02][13]	LED6, LED7
CPU Source [0] and CPU Source [9:5]	LED0, LED5, LED6, LED7, LED8[1)], LED9

1) Not available in MxL25641.

9.5.2.1 SSO Controller

The SSO controller targets to reduce the number of pins on the package to drive up to 32 outputs.

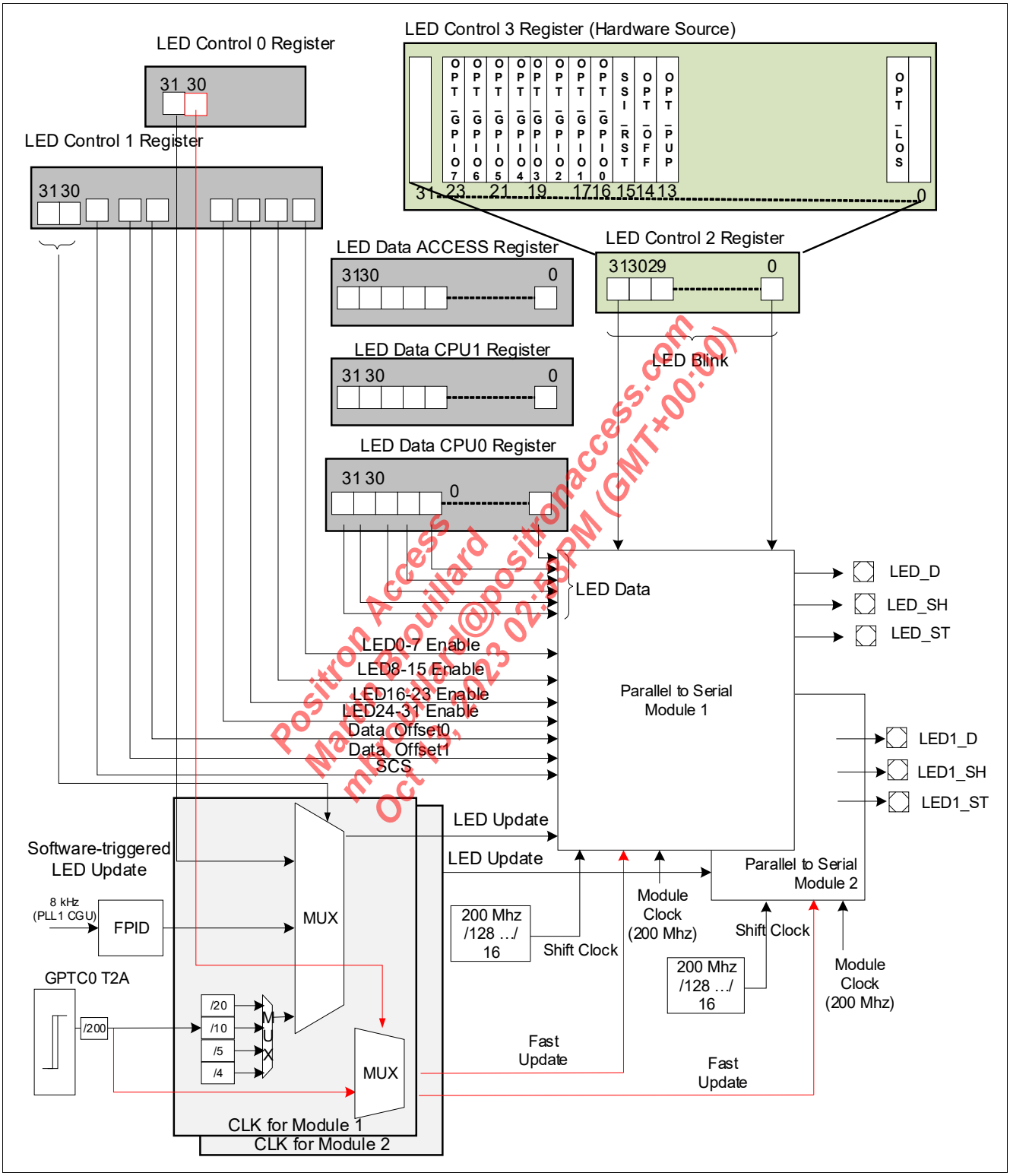


Figure 66 LED Controller Overview

Attention: MaxLinear recommends to use Fast Update mode for best LED blink results.

Serial Shift Output Update and Store

There are two options for updating.

- Single Update: The software must set the update bit in the control register to enable the hardware to start updating. After the update command, the hardware starts shifting out the data with the shift clock and at the end generate the store impulse.
- Periodical Update: The hardware is programmed to generate a periodical update. MaxLinear recommends to use the Fast Update mode.

The hardware must generate all data outputs according to the shift clock `LED_SH`. The `LED_SH` clock must have a 50/50 duty cycle. The internal update signal forces the SSO controller to start generating the `LED_SH` clock and LED_D. At the end of the programmed number of bits (8, 16, 24, or 32), the hardware generates the store pulse. This enables the store register in the external device to latch the data from the shift register. The outputs of the external shift register are always enabled. The software can write new data to the register at any time.

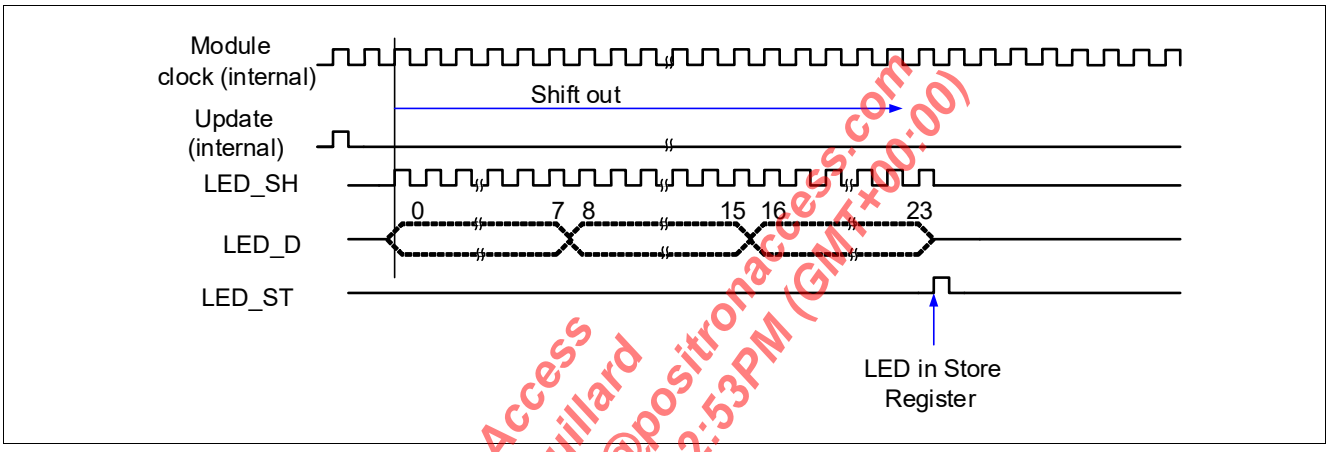


Figure 67 SSO Signal Generation - Example for 24 bits with Single Store Command

Figure 68 shows in more detail how the periodical update procedure works.

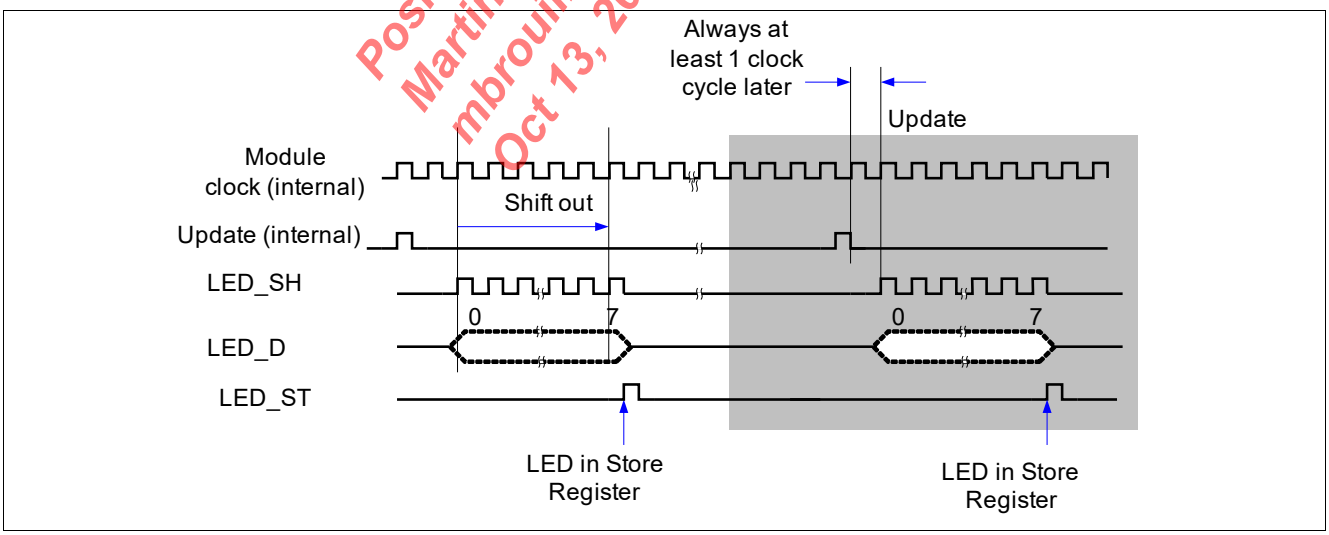


Figure 68 Periodical Update Procedure with 8-bit SSO Example

9.5.3 External Interface

The external interfaces have these external signals.

SSO with External SHIFT Register

The implementation of the SSO requires an external shift register.

Figure 69 shows an example of a typical type.

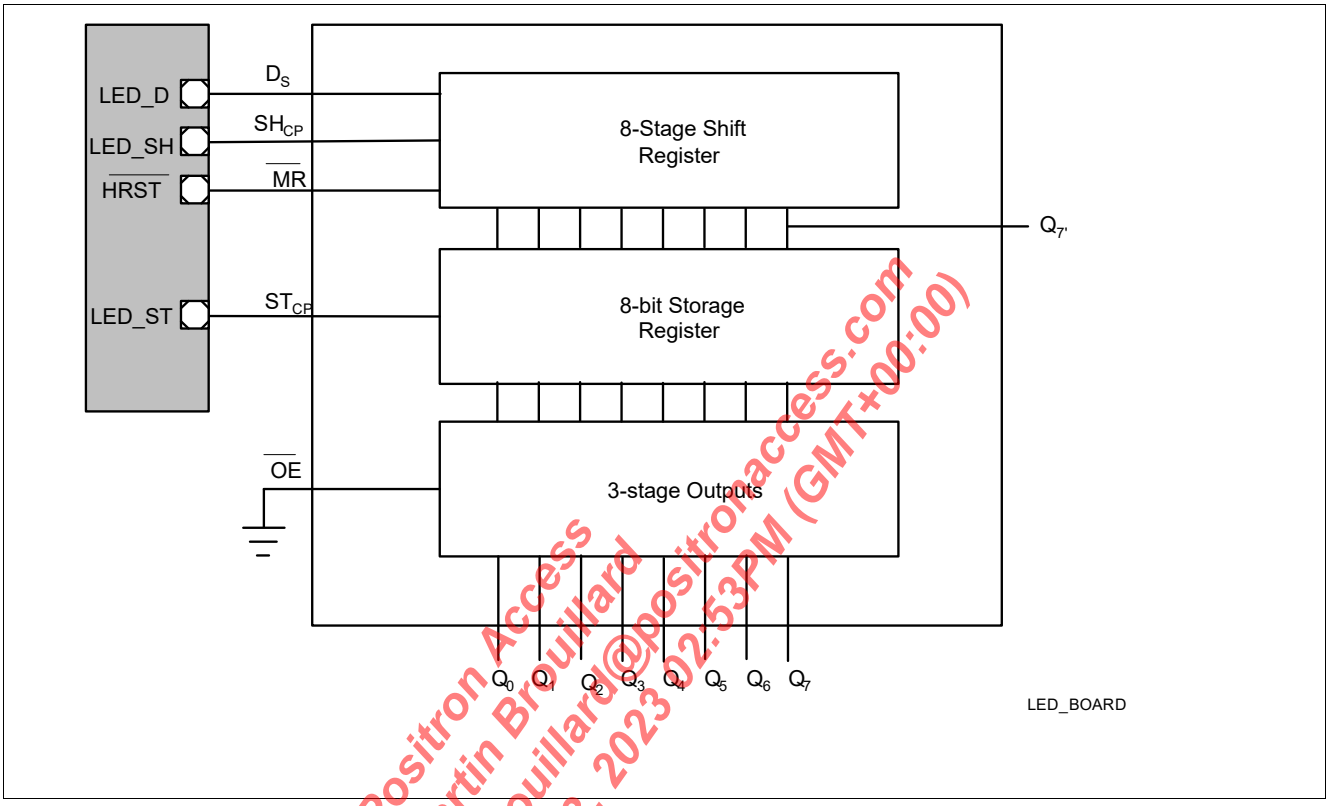


Figure 69 SSO Shift Register Connection Example with 74LV595

Brightness Detection Circuit

Figure 70 shows an example for a brightness detection circuit. Refer to [7] for more information.

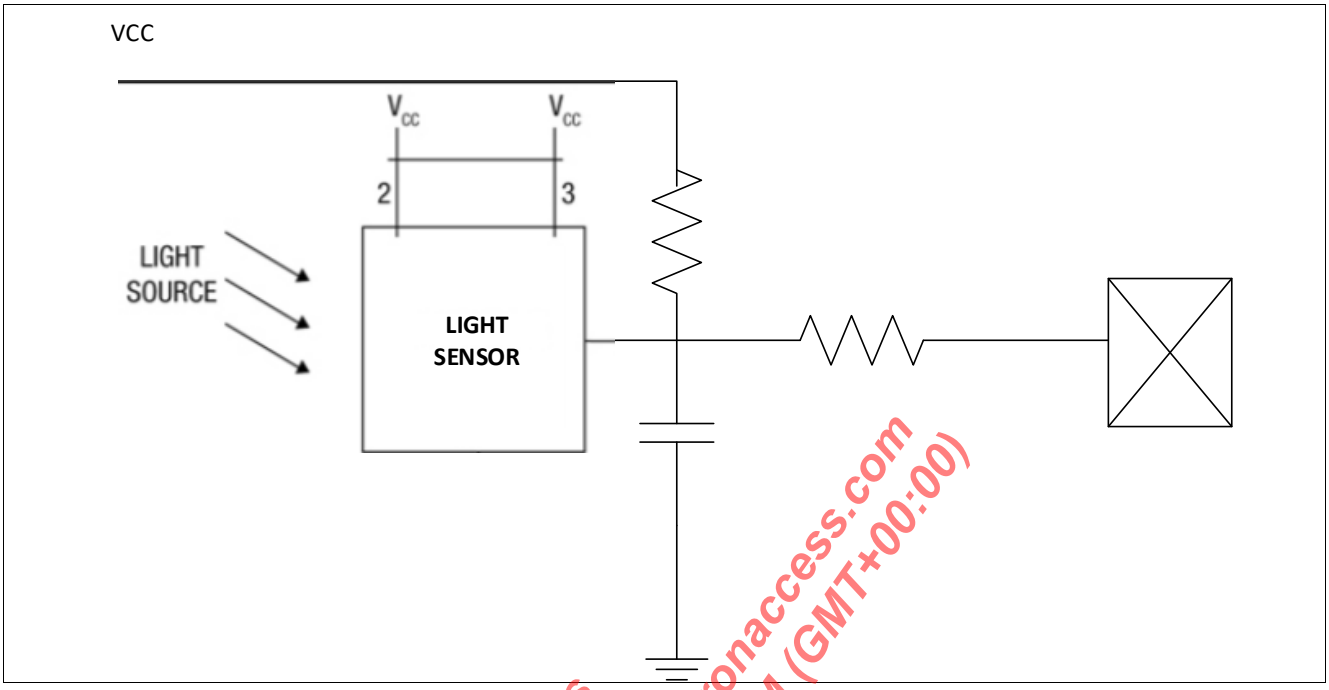


Figure 70 Brightness Detection

9.6 MDIO Controller

There are two Management Data Input/Output (MDIO) instances. Their supported features are:

- Two master MDIO interfaces, used to control internal and external Ethernet PHY/switch devices.
- External devices can be attached to either one of the master MDIO interface.
- Supports both Clause 22 and Clause 45.
- Automatic scanning of link status change in PHY devices
- Supports direct link parameters input (speed, duplex, flow control, EEE LPI, link status) from the Ethernet PCS Clause 37 and on-chip Ethernet PHY.
- The management agent can access the external PHY/switch or internal PHY through MDIO access.

9.6.1 MDIO Master Interface

The MDIO master module provides the register interface to access external or internal device registers. The access is triggered by an internal bus master access or by an automatic link status scanning function.

The interface uses the serial protocol defined by IEEE 802.3, Clause 22 or Clause 45. Up to 32 external devices can be addressed through a 5-bit PHY address (PHYADR). Each MDIO access includes start type (Clause 22 or Clause 45), access type (address/read/write/post-increment command), 5-bit PHY address, 5-bit register/device address and 16-bit read/write data

The MDIO_0 and MDIO_1 interfaces share pins with the GPIO function. When no external PHY or devices require access via the MDIO_0 or MDIO_1 interface, the pins are configured to GPIO or other alternative function.

9.6.1.1 High Speed Operation

The standard MDIO protocol uses a clock rate of around 2.5 MHz on the MDC. This is the default setting. To speed up the data exchange, it is possible to increase the clock generated on the MDC to a maximum of 25 MHz.

9.6.1.2 Automatic Polling State Machine

This helps reduce the load on the software for checking the devices status on each port. It also helps in automatic connectivity detection (hot-plug-in) of PHY devices. This command has the lowest priority and can be enabled to always run in the background.

This command performs read accesses to two specific registers (1 and 8) of each device on all ports to check their link and device statuses and to update the corresponding status registers. Auto-scanning for each link can be independently enabled. The Clause 22 or Clause 45 mode is programmable for each link. It is possible to enable each link on MDIO master interface 0 or MDIO master interface 1.

9.6.1.3 MDIO Master Indirect Access

When the MDIO master interface is accessed by the management action using the dedicated indirect MDIO register access. It has the higher priority over the auto-polling state machine.

9.7 I2S Controller Module

The Inter-IC Sound (I2S) bus was developed by Philips Semiconductors (now NXP Semiconductors). This specification is compliant with the I2S specification.

Both transmitter and receiver can generate clock. The role of generating clock is considered as system master. The word select line is driven by the system master to indicate the sound channel, `WS=0` is left channel and `WS=1` is right channel.

Data is sent by transmitter on rising or falling edge of the clock. However, the receiver always uses the rising edge to latch the data.

Features

- Two identical I2S controller are embedded.
- Supports master and slave mode.
- Compatible to the I2S specification
- Data transfer in adjustable audio sampling rate (48 kHz, 32 kHz, 24 kHz,16 kHz,12 kHz, and 8 kHz)
- TOPSPIN peripheral for synchronous/asynchronous operation of the interface, including AHB-BPI interface system integration, also including the FIFO concept to allow an unload of the bus system by buffering the data.
- Configurable word length (16, 18, 20, 24, 32, 48, 64, and 96 bits) and different frame length (192, 128, 64, 48, and 32 bits)
- Switch left/right channel order in frame
- Left/right alignment of words in frame
- Bi-directional transmission with independent receiver and transmitter

Generic Features of the I2S

This generic feature depends on the final implementation of each instance.

- Tx/Rx FIFO depth. Currently 8 x32 bit stages are supported.

9.8 PWM FAN Controller

The 4-wire Pulse Width Modulation (PWM) fan controller is integrated.

9.8.1 Features

The PWM controller supports:

- Intel 4-wire PWM controller fans specification revision 1.2
- PWM frequency nominal 25 kHz, with a range from 21 kHz to 28 kHz
- The PWM signal pin meets ViL<=0.8 V; less than 5 mA; Open Collector type.
- The PWM controller must set the fan speed to be in the range of less than 30% of FULL speed to 100% full speed.
- The PWM startup pulse resulting fan RPM must be less than 30% full speed and last less than 2 seconds.
- The PWM controller detects a fan locked state, or fan speed configured lower than the minimum fan speed, and reports to the system via interrupt.

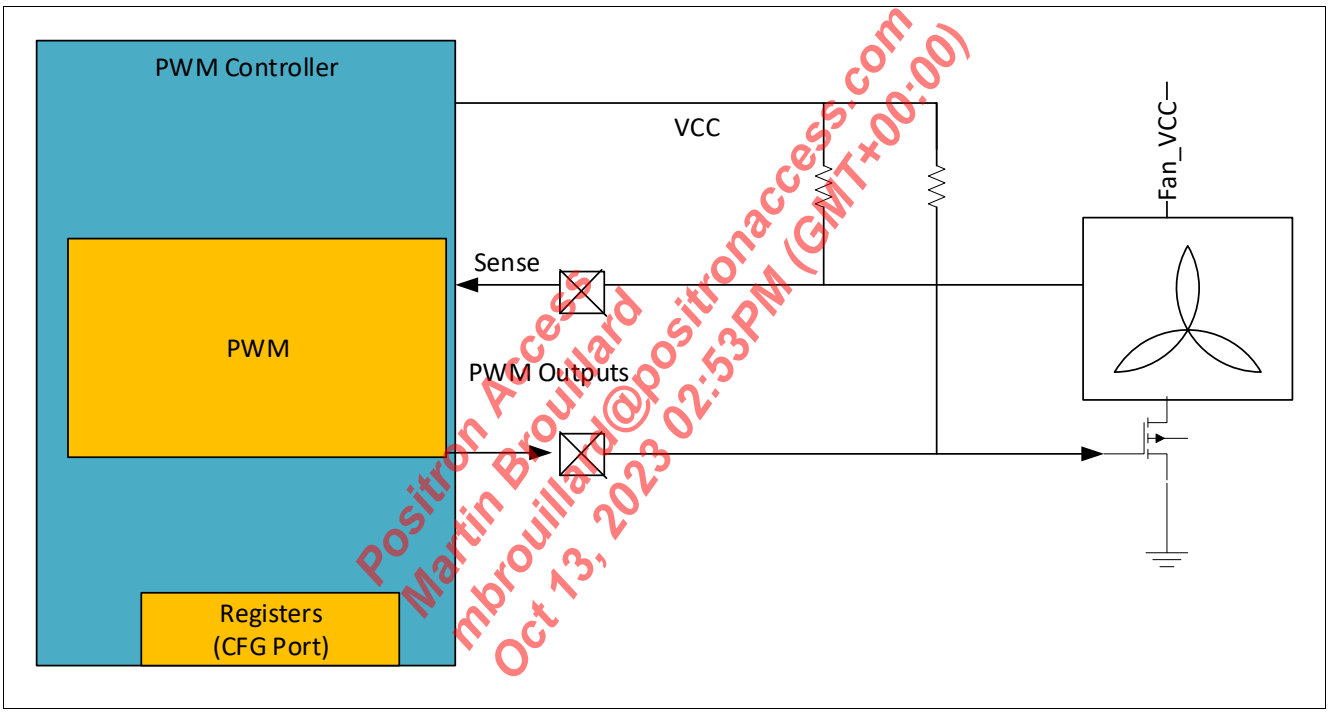


Figure 71 PWM Fan Controller Application Block Diagram Example

9.9 POR Module

The POR monitors the digital core voltage and generates the reset whenever the voltage level lower than the predefined level. The POR generates a low active pulse and maintains low active for at least 80 ms.

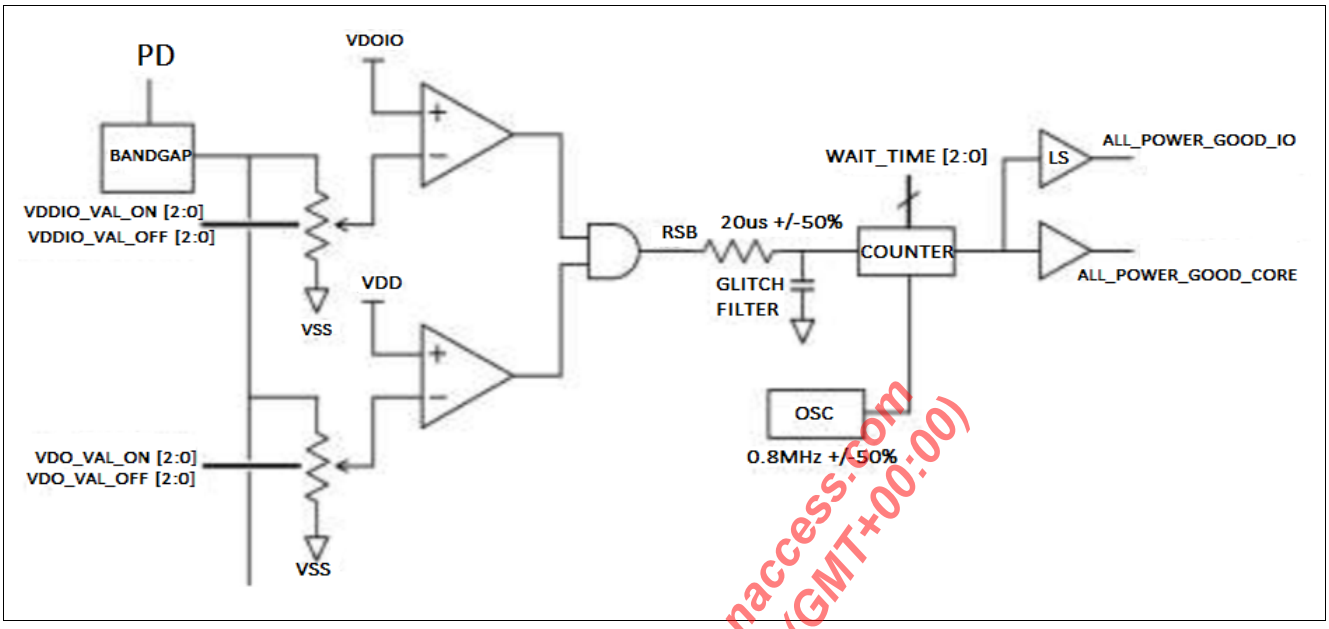


Figure 72 Block Diagram of POR

Attention: The ON trigger voltage is set to 0.66 V and the OFF trigger voltage is set to 0.60 V. The WAIT_TIME is set to 81.92 ms.

9.10 Overcurrent Detection Comparator

There are two comparators for overvoltage/overcurrent detection for USB host or PON dying gasp application.

The comparators come with programmable hysteresis feature.

The comparator output is connected to IOAPIC, in the same time routed to the GPIO pins.

USB_OC_INT[1:0] are connected to **IO01_SPI2CS2** and **IO02_SPI1CS4**, to allow direct use of the on-chip detection circuit for controlling the external USB supply (for example MOSFET).

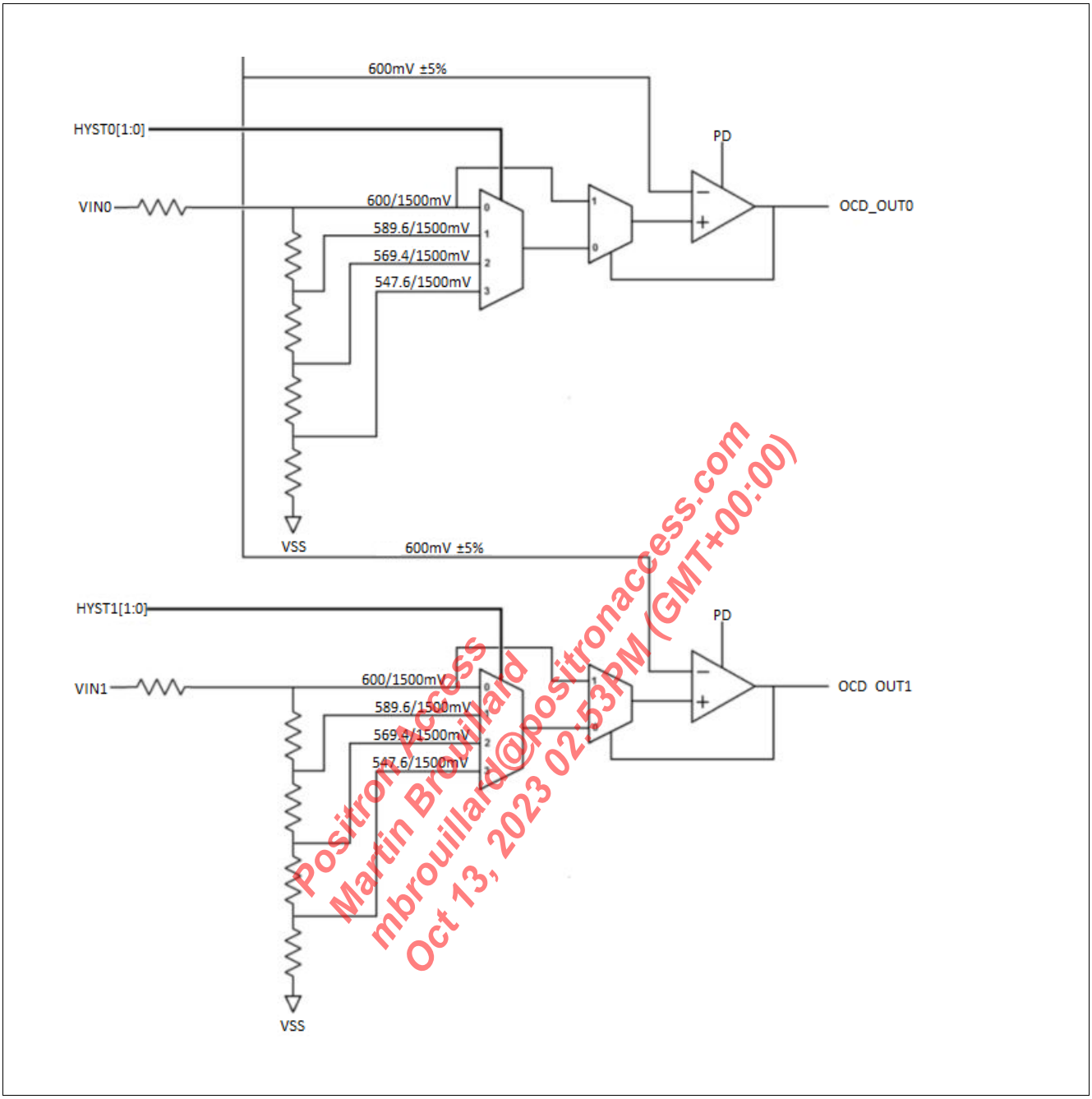


Figure 73 Overvoltage Detection Circuit

Note: The 600 mV +-5% reference comes from on-chip internal bandgap reference.

Figure 74 is an example of system application.

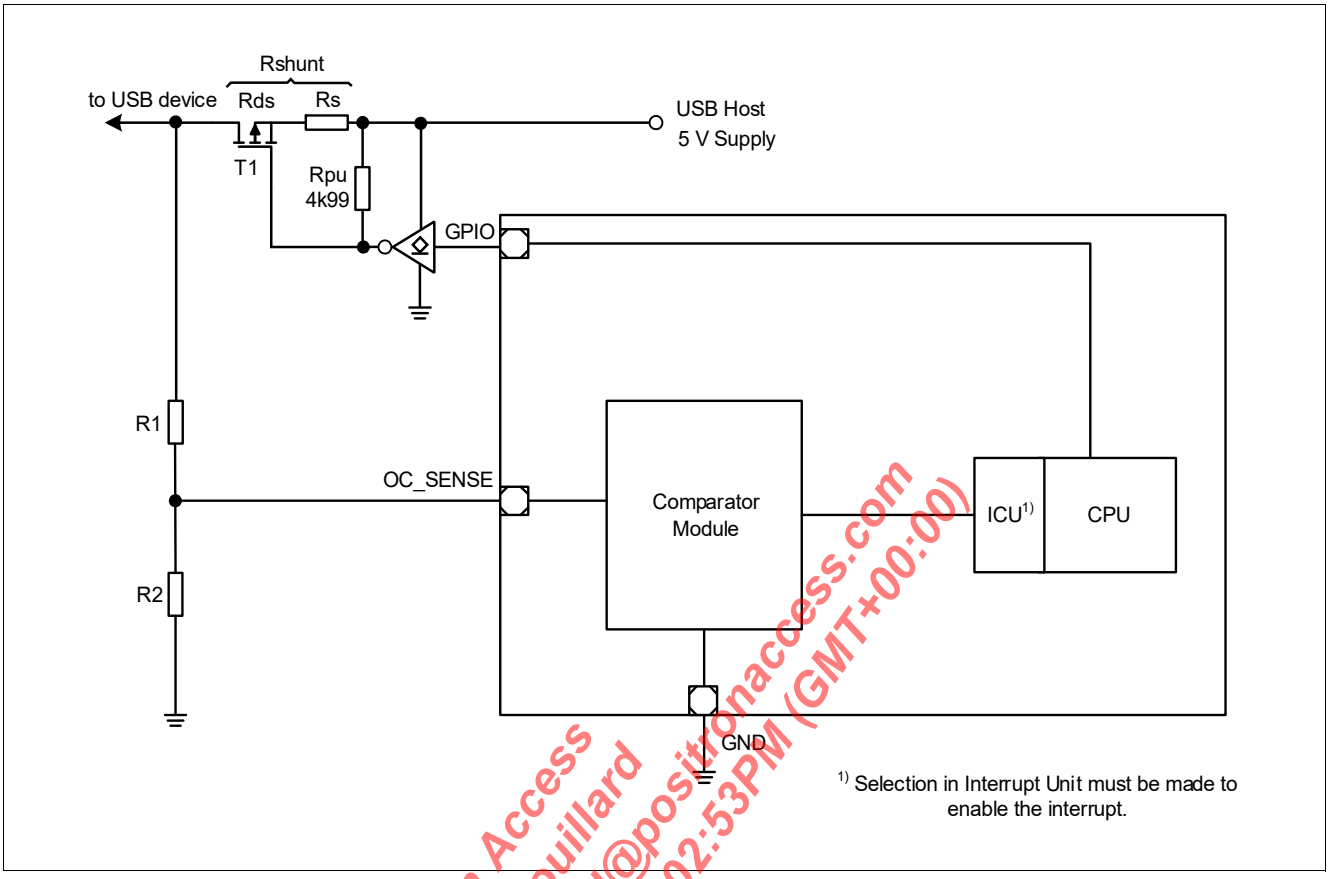


Figure 74 USB Overcurrent Protection

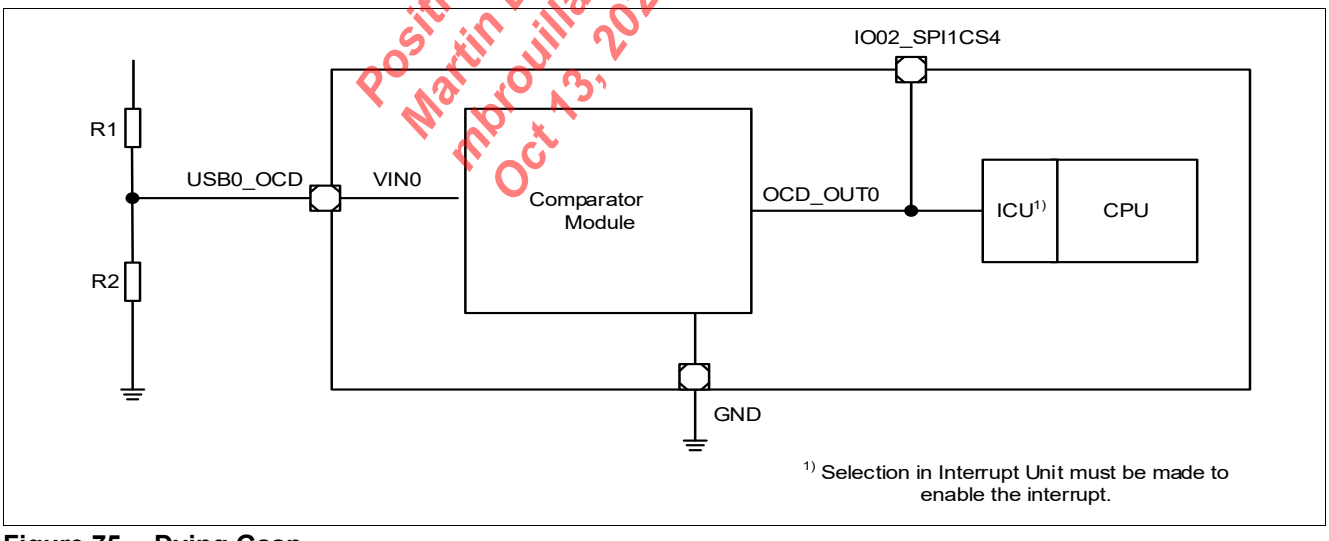


Figure 75 Dying Gasp

The dying gasp monitors the external input pin (**USB0_OCD**) (Connected to `VIN0` via an Analog Input PAD) and generates the interrupt whenever the voltage level lower than the predefined level. The `OCD_OUT0` signal is feed to interrupt **USB0_OC** as well as **OCD_OUT0**.

`HYST0` is configurable from register.

Resistor Calculation and Selection

The shunt resistor Rs and MOSFET T1 must be selected. For the targeted maximum current I_{max}, the desired voltage drop Vdrop occurs.

$$Rshunt = \frac{Vdrop}{Imax} \tag{3}$$

For V_{drop} = 250 mV and I_{max} = 900 mA, the total shunt resistor R_{shunt} equals 278 mΩ.

It is a good design practice to select a MOSFET T1 which has significant lower R_{dson} than R_{shunt}, because R_{dson} of MOSFET varies a lot over temperature and also from part to part.

Selecting a MOSFET T1 with R_{dson} = 40 mΩ and a shunt resistor of 240 mΩ is a good choice.

The external resistor divider R1/R2 is used to set the detection point Vref (= 1.5 V). Around that point, a hysteresis can be programmed.

Assuming a quiescent current Iq of about 0.75 mA through R1 and R2 with current into the OC_Sense pin negligible, a switching point V_{drop} (= 250 mV) below the USB power supply voltage V_{usb} (= 5.0 V) the resistor divider R1/R2 can be calculated as follows:

$$R1 + R2 \sim \frac{Vusb - Vdrop}{Iq} = \frac{5.0\ V - 0.25\ V}{0.75\ mA} = 6.33\ kOhms \tag{4}$$

$$Vref = \frac{R2}{R1+R2} * (Vusb - Vdrop) \tag{5}$$

$$\frac{R1}{R2} = \frac{Vusb - Vdrop}{Vref} - 1 = \frac{4.75\ V}{1.5\ V} - 1 = 2.1667 \tag{6}$$

With (5) and (6), the resistors can be selected as follows:

$$R2 = \frac{Vref}{Vusb - Vdrop} * (R1 + R2) = \frac{1.50\ V}{4.75\ V} * 6.33\ kOhms = 2\ kOhms$$

R1 = 4.33 kOhms

(7)

R2 is available in the E96 series, the closest resistance for R1 in the E96 series is 4.32 kΩ, therefore the final selected values are:

R1 = 4.32 kΩ, 1%

R2 = 2 kΩ, 1%

The R1/R2 ratio is more important than the sum of resistances (R1 + R2). In case other overcurrent conditions are selected and neither R1 nor R2 are calculated to be close to an available resistor value, it is recommended to select R1 and R2 to meet the ratio as close as possible from the E48 or E96 resistor series and use (R1 + R2) of 6.33 kΩ to be met within targeted tolerance.

Tunable Hysteresis

The hysteresis allows to adjust the range of detection around the targeted switching point. The hysteresis is programmed by the software driver.

9.11 Temperature Sensor

A temperature sensor with four sensor elements to measure different hot spots is integrated in the SoC.

The temperature sensor generates 12 bits output to indicate the chip junction temperature. The accuracy of the temperature sensor is +/-3°C.

Equation (8) shows the mapping between temperature and the sensor output.

An over-temperature detection logic is built into the SoC. The over-temperature threshold code can be programmed - the default value is Tj = 110°C. When the chip junction temperature is over this threshold, an over-temperature interrupt is generated to the CPU.

$$Temp_{mode2}(C) = G + H * Eq_{bs} + J * F_{clkm}$$

Where:

$$Eq_{bs} = \frac{N_{bs}}{cal5} - 0.5$$

$$F_{clkm} = F_{clk}\ (in\ MHz)$$

Where

- Temp = Temperature in degrees centigrade
- N_{BS} = Output (digital output, d_{out}, latched after conversion)
- For values of G, H & cal5, see electrical specifications

(8)

Attention: Mode2 is the temperature operational mode.

Attention: Nominal value for G,H, J, cal5 are 60,200,-0.1,4094; Fclk is 4 MHz.

9.12 On-Chip Packet Buffer

The integrated on-chip packet buffer size is 512 Kbytes.

The complete memory or a part of memory can be used for data packet storage. The on-chip packet buffer is managed by Free Segment Queue Manager (FSQM) inside the Buffer manager (BM).

9.13 Voice Subsystem

This section defines the voice subsystem.

9.13.1 Voice Subsystem

URX851, URX850, and MxL25641 inherited the voice subsystem of Puma™ 7 SoC. The voice subsystem offers these external interfaces:

- 2x TDM
- 2x SSI

9.13.1.1 Voice Subsystem Overview

The main features of the voice subsystem are:

- A DSP Core
- L1 SARAM - Banked Memory
  - Provides DARAM replacement
  - SARAM total size is 32 Kbytes, 16 banks
- L1 Data Cache
  - 2-way set associative, with 4 KW each way, totaling 8 KW
  - Software allocation of cache memories to RAM
- L1 Instruction Cache
  - 2-way set associative, with 4 KW each way, totaling 8 KW
  - Software allocation of cache memories to RAM
- L1 Default MMU Cache Policies
- L2 MMU with Additional Configurable Cache Regions
- L1 System Interface
- 22 Incoming Interrupts
- Idle/Full-idle
- Peripherals to Support Voice Flow:
  - 2x TDM
  - 2x VCODEC
  - DSP INTC
  - Clock and reset control units
  - 3x Timers
  - DSP proxy
  - Boot and configuration

The interfaces to the SoC are:

- OCP Master
- OCP Slave
- Clock Input from the SoC
- Reset Input from the SoC
- Pin Muxing with Peripheral Interface to SoC

Table 127 lists the supported TDM modes.

Table 127 TDM Modes Supported

	TDM	CODEC/SPI mode0	SSI0 IF	SSI1 IF
TDM0	x	x		
TDM1	x	x		
TDM2			x	x

9.13.1.2 TDM and SPI

The TDM1 interface represents the PCM interface with FSC and PCLK and is used to connect DXS102 or DXS101. It is also possible to connect other devices to TDM1, such as DECT parts.

The TDM2 interface is a chip internal interface to the VOCODEC which provides the x2 SSI interfaces.

These modes are available:

- Up to 4x FXS with SLC220 using the SSI interface
- Up to 8x FXS with DXS102 using the TDM1 interface

9.14 SDXC Interface Controller

The SDXC card host is integrated in URX851 and URX850, but not in MxL25641.

The SDXC supports:

- Part A2 SD Host Controller Version 4.10
- Part E1 SDIO Specification Version 4.10, dated April 29, 2014
- AMBA, AHB Specification Version 2.0
- AMBA, AXI Specification Version 1.0

Note: UHS-II is not supported.

These are the SDXC features.

- Host clock rate variable between 0 - 104 MHz
- Up to 416 Mbps data rate using 4 parallel lines
- 1-bit and 4-bit SD modes
- SDR50 and DDR50 modes
- Up to 2 TB card density
- Built-in DMA with up to 2 Kbytes block buffer (1 Kbytes block size)
- Implements tuning block for SDR104 mode for programmable delay up to 32 delay taps
- Cyclic redundancy check CRC7 for command and CRC16 for data integrity
- SD Host Controller Spec (Version 3.0) register set
- Variable-length data transfer
- Performs READ wait control, suspend/resume operation SDIO card
- Designed to work with I/O cards, read-only cards and read/write cards
- Supports read wait control, suspend/resume operation
- Card detection (insertion/removal) via SD_CD# pin

10 Security

The security description covers these sections:

- **Trusted Execution Environment (TEE)** (**Section 10.1**)
- **Secure Platform Engine** (**Section 10.2**)
- **DDR Content Protection with In-line Encryption Device Engine** (**Section 10.3**)
- **PCIe Inline Protection** (**Section 10.5**)
- **Crypto Engine** (**Section 10.4**)
- **Multi Processor System** (**Section 10.6**)

10.1 Trusted Execution Environment (TEE)

This section describes the TEE aspect from hardware design, including the CPU core which is targeted to run the secure OS and software.

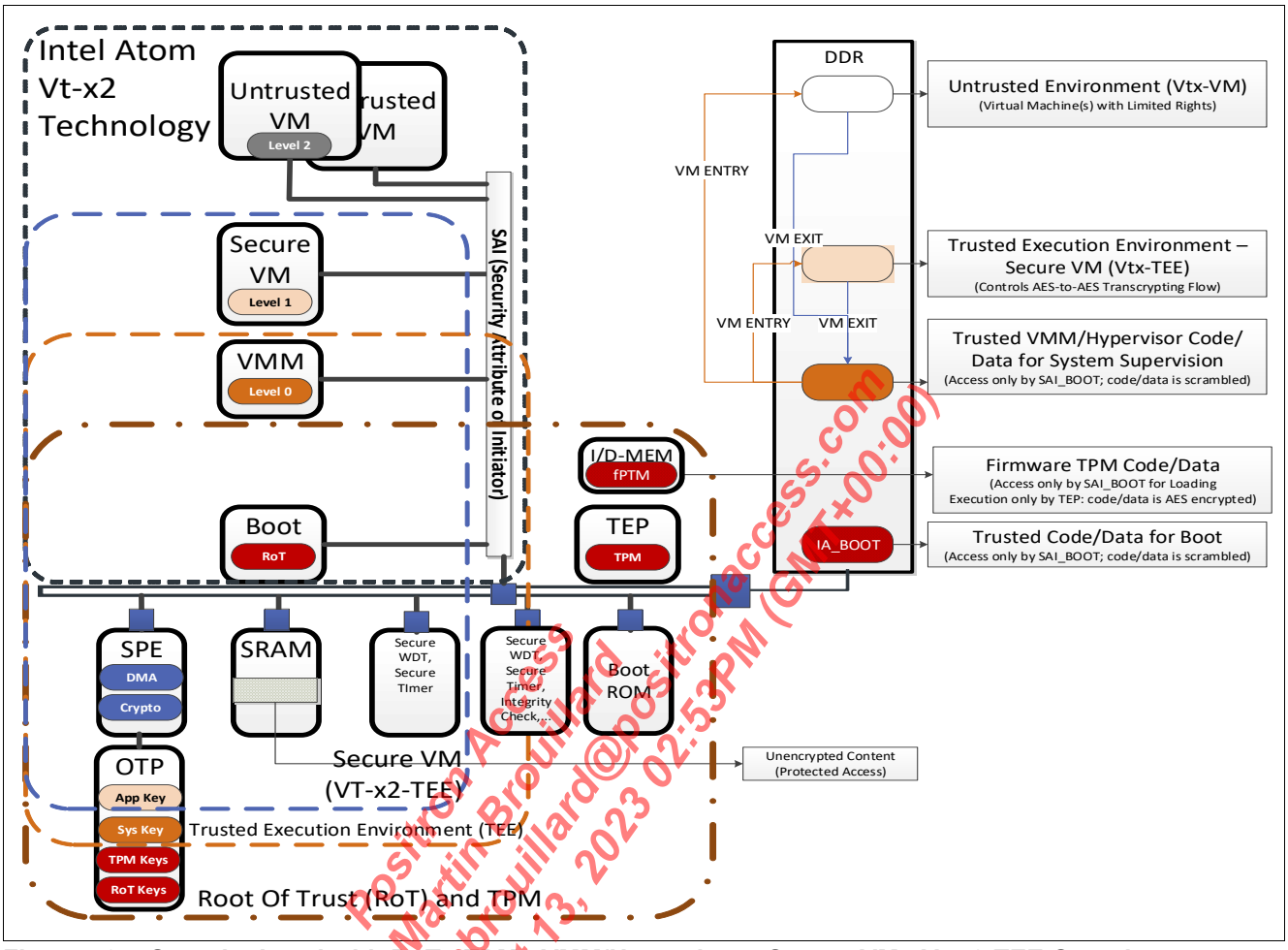


Figure 76 Security Level with RoT - TPM - VMM/Hypervisor - Secure VM - Vt-x2-TEE Overview

Attention: Although TPM is a possible application of TEE from a hardware design point as stated in Figure 76, there is currently no plan to enable a system solution for TPM.

10.1.1 Features

The TEE provides these features:

- **NoC Access Control Security**
  - Implement security and fire-walling functionality in each target agent via protect region
  - Protect region access permission set-up is done by TEP only
- **Secure Platform Engine** (SPE), trusted application execution, secure storage
  - OTP integration supported by exclusive READ access of SPE
- **NoC Access Control Security** allow to setup access controls from initiators (NoC masters)
- **Multi Processor System** to provide semaphore based mailbox system between Intel Atom and TEP core.
- **Protected JTAG** to prevent debug of processors in deployed products
- **Vt-x2 SAI Control** to support isolated software environments on the Intel Atom VMs software.

10.1.2 Trusted Execution Processor

The Trusted Execution Processor (TEP) is added to address functionality:

- Measured boot
- Attestation and integrity check on the **Intel Atom CPU Subsystem** uses DDR hash when returning from idle
- Trusted Platform Module (TPM) capabilities but not the entire TPM specification is targeted by URX851/URX850/MxL25641.
- Watchdog timer pre-warning reset request to allow capture of reset cause.

Consult the system software description for details on which features are supported by the system software.

10.1.3 Access to Platform Resources

The platform resources are described in these sections.

10.1.3.1 Root of Trust

The root of trust is stored inside the OTP. Access to the OTP is limited by access `hw_id` and software IDs. Only the Intel Atom `IA_BOOT` and TEP core have crypto officer privilege on SPE.

10.1.3.2 Access Protection

The access protection to other on chip/off chip resource is achieved via access initiator to address region mapping.

Each access initiator is assigned a dedicated initiator ID. Based on this ID, the interconnect allows/stops access to certain address regions. The ID to region mapping is defined in the interconnect register, and these registers are only allowed to be set-up or changed by the `IA_BOOT` or TEP. These environments can give access to the next level, `IA_VMM`, but must explicitly do so in the access control register.

10.1.3.3 Vt-x2 SAI Control

In the Intel Atom subsystem, the access control behaves differently for cached and uncached L2 but cached L1 access to interconnect:

- Cached access, the EPT regions are compared to guarantee access rights;
- Uncached access, the Security Attributes of Initiator (SAI) tag is maintained.

For cached and uncached access, one fixed SAI value is used at NoC level to give access to DDR via the DDR firewall.

10.1.3.4 NoC Protection Mechanism Overview

The NoC allows access to specified Target Agents (TA) to be restricted, protecting these TAs and their connected system slaves from unauthorized access. The protection mechanism is based on the use of flexible, pre-programmed protection regions that enable fine-grained access control at the TA, based on these attributes for each request.

- SAI Bits Setup
- NoC IA Access Rights Register (to be programmed by TEP/VMM only; eight groups for URX851/URX850/MxL25641)

10.1.3.5 NoC Access Control from Software

Setting up the protection region register can by default only be performed by the `IA_BOOT` SAI of Intel Atom and TEP. This is controlled by setting up the NoC registers, see **NoC Access Control Security** for more details.

10.1.3.6 TEP Core Subsystem

The **Trusted Execution Processor** (TEP) has several top-level security relevant capabilities.

10.1.3.6.1 TEP Watchdog

The watchdog part of the TEP processor provides standard watchdog functionality.

10.1.3.6.2 TEP Bus Privilege Levels on external Buses

On the buses outside the CPU, AHB5 sideband signals can indicate whether transfers originate from non-secure world (NS bit), and additionally which SID is associated with them.

10.1.3.6.3 TEP Parity Integrity Protection

The parity integrity-protection feature implement configurable error detection on ICCM, DCCM, and several critical internal TEP processor registers. Parity adds either one bit for each 32-bit word or one bit for each byte. Parity mismatches trigger an exception. The integrity-protection feature adds protection against FI attacks.

10.1.3.6.4 TEP Closely Coupled Memories

The CCMs are memories that are attached to dedicated buses within the processor core. They consist of instruction CCM and/or data CCM. With SecureShield, ICCM and DCCM can marked as fully secure or non-secure, or with a 50% split. The CCM itself restricts access based on the access bit present in the extra access-control data on the bus, set by the CPU or debugger.

10.1.3.6.5 TEP Code Protection

The code-protection option is intended to block loads and stores to individual memory regions. The memory is partitioned into 16 equally sized regions covering the entire memory. Each of the memory regions is associated with an external protection bit, held in a fuse or register. This protection may also be applied to MPU protections; for a read or write to succeed, both the MPU and code protection must allow the access.

10.1.3.6.6 TEP Stack Checking

Stack checking is a memory-protection option that checks that stack accesses are within specific bounds. These bounds can be configured separately for kernel space and user space, and for SecureShield these bounds are duplicated for the secure and normal worlds. Access to the configuration registers is limited to kernel space. Stack accesses outside these bounds generate an `EV_ProtV` exception.

10.2 Secure Platform Engine

The selected trusted platform solution is based on a secure module named Secure Platform Engine (SPE).

10.2.1 Block Diagram of SPE Integration

Figure 77 shows all supported interface in separated groups to give an indication of the default usage.

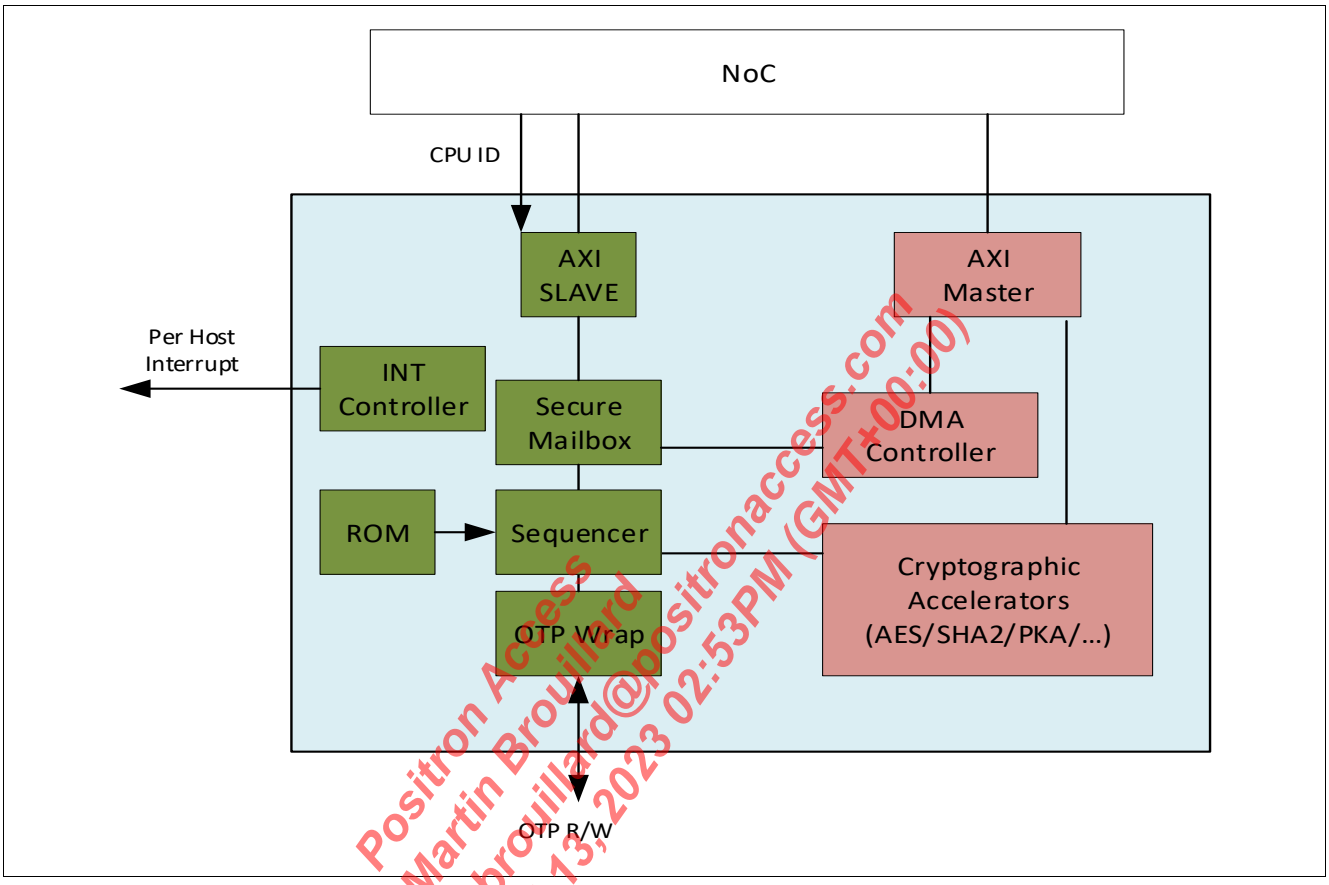


Figure 77 SPE Overview Block Diagram

10.2.1.1 Features

The trusted platform solution is designed to protect sensitive data, protect embedded system against hackers of trying to reinstall hacked OS, copy hacked software image to other system, replace critical applications, and protect device manufacturer's software against illegal copying.

The support trusted solutions are:

FIPS 140-2 level 2 Security Features

- On-chip NVM as root of trust
- Cryptographic Algorithm Validation Program (CAVP):
  - RSA, ECDSA
  - DRGB
  - Triple-DES (ECB, CBC)
  - AES (ECB, CBC, CTR, XTS)
  - AES-CCM, AES-CMAC
  - HMAC, SHS for SHA-1, and SHA-224/256/512
  - KDF

Asset Store

- Ownership keeps assets from different applications and different hosts separated:
  - CPU identity: 2 bits and 1-bit normal/secure world
  - Application identity: 32 bits
  - Application long-term identity: 33 up to 224 bytes
- Number of concurrent assets supported scales with the crypto module RAM size.
- Root key based security via OTP interface
- Support of symmetric and asymmetric key storage and export using AES SIV key-blobs
- Symmetric key generation and derivation using NIST SP 800-108 algorithm
  - Derive a KDK from a root key
  - Derive a KEK from a root key
  - Derive a key asset from a KDK
- Shared secret generation based on asymmetric key exchange
- KDF NIST SP 800-56A and B – HMAC and CMAC-based
- Key derivation according to NIST SP 800-135. Authentication to allow access control to assets

Timers

- Time stamping of keys
- Key lifetime and access window management based on time stamp
- Functions are based on hardware timer in a dedicated clock domain

Power Management

- Support for sleep mode
- Support for hibernate mode
- Automatic clock switching logic for low power consumption when operational
  - Clock switching per cryptographic sub-module and clock domain

FIPS 140-2 and 3/SP 800-90-compliant True Random Number Generator (TRNG)

- Hardware-based, non-deterministic RNG
- Used to internally generate session keys, IV's, nonces, cookies, public and private keys
- RNG using SP 800-90 NRBG, SHA-2 conditioning, and AES-256 CTR_DRBG

Public Key Accelerator

- Hardware-accelerated public key operations for secure boot using ECDSA-256/384/521
- Signature sign and verify using RSA-PSS, RSA-PKCS#1.5, and ECDSA
- Basic operations for DH, RSA, DSA, and ECDSA, including:
  - ECDSA-256 and ECDSA-384/521,
  - ECDH
  - RSA-1024, RSA-2048, RSA-3072, and Curve25519 support

Crypto Algorithm

- DES, TDES
  - ECB, CBC
- AES
  - ECB, CBC, CTR, ICM, f8, XTS
  - CMAC, CCM, GCM
  - Wrap and unwrap (NIST SP800-38F)
  - Key lengths: 128, 192, 256, 384, and 512-bit

Hash Algorithm

- Basic hash and HMAC modes
  - SHA-1,
  - SHA-224 and SHA-256
  - MD5
  - SHA-384 and SHA-512

Management of Non-volatile Assets

- Monotonic counters:
  - Short monotonic counters can be stored internally in OTP (up to 127 states), these can be used for host firmware version numbering or for other infrequently updated counters
  - Long monotonic counter have to be stored externally in non-volatile system memory.
- Private to the crypto module, read and write access
- OTP memory management and access:
  - Static Asset (private to Asset Store)
  - Authentication Key Asset (private to Asset Store)
  - Trusted firmware version and other short nonvolatile counters
  - Public data (can be requested via the token interface)
  - 16 k bits OTP for asset store
- Trusted firmware version management by (short) monotonic counters
- OTP write protection by OTP status bits
- Partially programmed in the factory during device production

Token API

- Token interface using mailboxes: command/response
- OTP memory usage (read and -protected- write):
  - Hardware Unique Key (HUK)
  - Provisioning, ROTPK, EK, SSK write
  - Authenticated firmware version update and validation
- Token API provisioning
- RNG, for configuration and random number generation
- Public key acceleration, including ECDSA and DSA
- Authenticated secure debug
- Asset management
- Autonomous data transfer using DMA
- Basic encrypt/decrypt and MAC calculation
- Hash and HMAC operations
- AES key wrap
- System information
- System abort request (soft reset)
- Output mailbox zeroization

Secure Debug

- Enables host system level secure debugging
- Bits can be enabled/disabled under control of public key cryptography

Secure Boot

- Image decryption
  - AES-CBC or AES-CTR depending on use case
- Image signature verification
  - ECDSA NIST P-256/384/521
  - SHA-256/384/512
- Helper functions
  - Symmetric key generation
  - Key derivation
  - AES wrap

Embedded Controller

- The ROM-based firmware program memory ensures system security.

Host Command Interface

- Token-based command interface
- Four separate mailboxes for commands
- Customer configurable for up to 4 separate hosts. The default is 2.
- Hardware separation. Each host can only access commands or results that are meant for that specific Host
- Normal (non-secure) and secure world commands and result are separated and cannot be accessed by the other world
- 32-bit AXI slave interfaces available

10.3 DDR Content Protection with In-line Encryption Device Engine

This section describes the IED engine for DDR contents protection.

This engine is available in URX850 by default. By default this feature is disabled by fusing in URX851 and MxL25641. Contact your MaxLinear representative for further information about how to order URX851 and MxL25641 devices with this feature enabled.

10.3.1 IED Integration and Features

This IP block is integrated into the DDR NoC subsystem.

The supported DDR contents protection features are:

- XTS-AES encryption/decryption for pre-defined DDR region; depends on the control register bit, DDR can be partitioned into: insecure region|| IED secure region|| insecure region; or IED secure region|| insecure region || IED secure region.
- 128-bit AES algorithm
- Up to 32 outstanding commands to be supported
- Ordering per AXI bus protocol
- Supports burst up to 256 bytes
- All traffic to be encrypted must be 16B size aligned, must feed the IED engine with multiple of 16 bytes data
- Configuration IED IP via standard AHB bus
- Global bypass for IED engine (zero latency path)
- Address based IED/non-IED dynamical choice of data path
- For data passing IED, programmable IED exclusion range (same latency with IED enabled)
- No limitation on the maximum size of encrypted data, although SP800-38E limits it to 2 power of 20 AES blocks
- Address up to 16 GB DDR

10.4 Crypto Engine

The crypto engine is selected for high speed security applications such as IPsec, SRTP, SSL, TLS, DTLS, and MACsec. This engine is available in URX851, URX850, and MxL25641.

10.4.1 Features

The crypto engine is specified in:

- High IPsec performance
  - 10G/5 Gbps for large/small packets for URX851 and URX850
  - 5G/2.5 Gbps for large/small packets for MxL25641
- IPsec classification
- IPsec transformation
- SSLv3.0/TLSv1.0/TLSv1.1/TLSv1.2
- TLS v 1.3 support with
  - Reduce handshake latency, 0 RTT;
  - Record payload protection cryptographic mechanism updates
  - New cipher type as Chacha20/Poly1305 in addition to AESGCM
  - New signature algorithm such as ed25519 and ed448
- DTLSv.0-v2.0 per RFC 4347 an 6347
- MACsec based on IEEE 802.1AE-2006 and 802.1AEbn
- SRTP per RFC3711
- Crypto engine: DES/3DES in ECB, CBC with 56-bit key/3x 56-bit key; AES in ECB, CBC, ICM, CTR with 128/192/256 bit keys, GCM, GMAC, and CCM modes
- ARC4 in stateful and stateless mode, up to 128 bit key
- Hash engine: MD5 and SHA-1; SHA-2 with 224 bit, 256 bit digest; HMAC; GHASH for GCM and GMAC; AES-CMAM, XCBC-MAC, and CBC-MAC
- Interface PRNG for optimal IV generation
- High performance AXI master bus: 128 bits with 36-bit address
- Multiple descriptor rings with individual access for multi-processor support
- Scatter/gather DMA
- Four ring managers and four interrupt controllers
- 8-bit VMID tag is driven out on sidebands signals of the AXI master interface to tag each data transfer Descriptor fetch, lookup table fetch, transform record fetch and update, flow record fetch and update, look-aside data fetch and store.
- Export control: This module can be disabled by production fuse to comply to certain export control requirements. Once the fuse bit is set, the whole module is not functional.

10.5 PCIe Inline Protection

This section describes the PCIe inline protection solution for e-t-e Root Complex (RC) to End Point (EP) product supported in URX851, URX850, and MxL25641.

Table 128 PCIe Terminology Abbreviations

Acronym	Description
ASLR	Address Space Layout Randomization
DAC	Discretionary Access Control
EP	End Point
EPT	Extended Page Tables
LXC	Linux Containers
MITM	Man-in-the-middle
PIP	PCIe Inline Protection
PTM	Precision Time Measurement
PIP	PCIe Inline Protection
RC	Root Complex
TEE	Trusted Execution Environment
TLP	Transaction Level Packet
TRNG	True Random Number Generator
VMI	Vendor Message Interface

10.5.1 Overview

The PCIe inline protection aims to provide mitigations against physical attacks. This section summarizes the features and software/hardware partitioning.

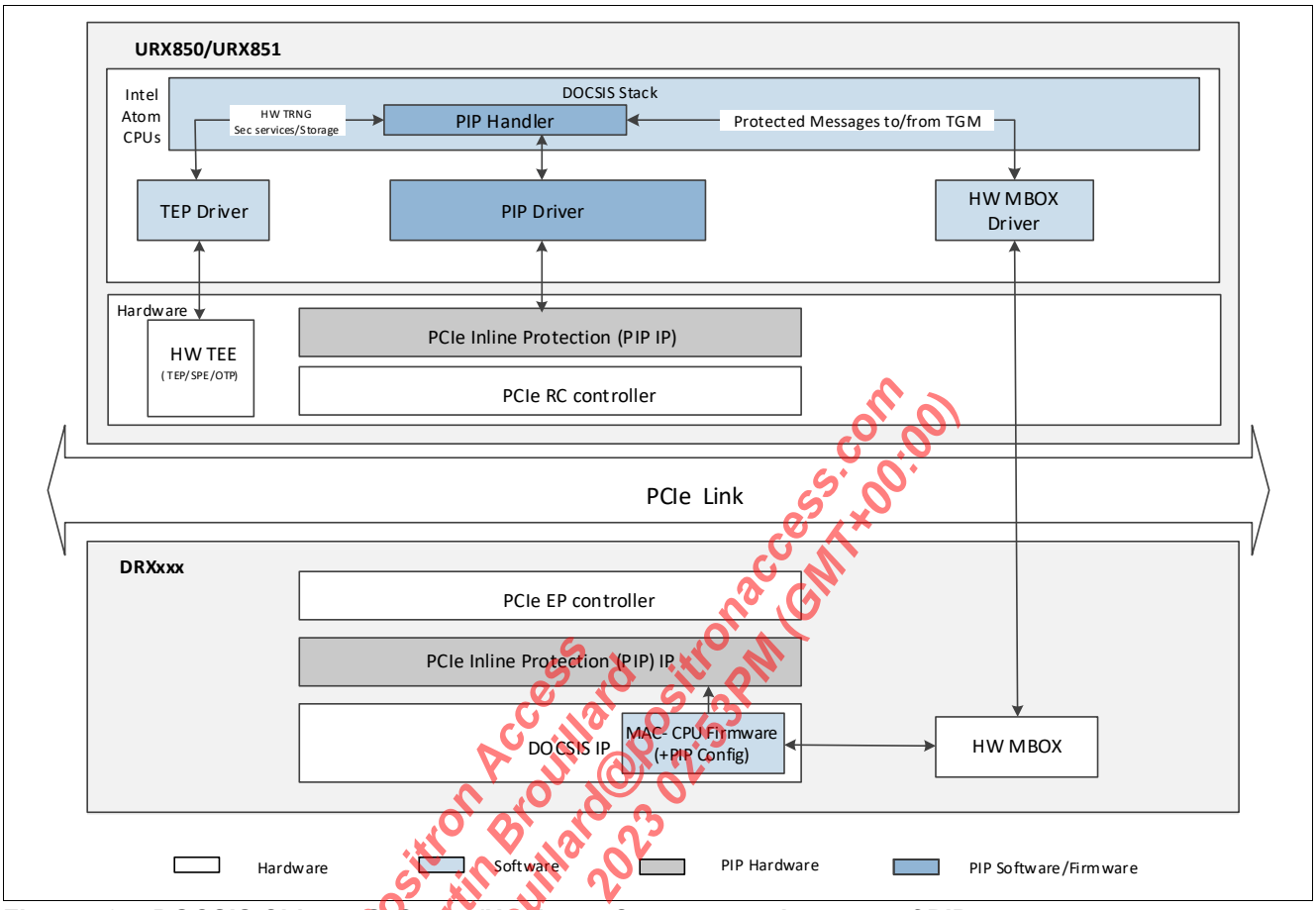


Figure 78 DOCSIS Chipset Software/Hardware Components in scope of PIP

The features of the PCIe inline protection are:

- PCIe Bus Tampering Detection
  - Detection of more than 1 PTM message sent during one measurement
  - Measurement in PCIe 4.0, 3.0, 2.0, 1.0 modes in either single or dual lane configuration
  - PTM message body contains 64-bit Security ID (per direction) pre-configured per PTM transaction.
  - URX851/URX850/MxL25641 service software module initiates PTM message latency measurement
  - PCIe EP hardware based PTM RC message match compare and auto-respond with PTM EP message is implemented to reduce measurement response to fixed deterministic hardware delay.
- PCIe TLP Packet Confidentiality
  - AES-CTR crypto algorithm
  - Reference software uses TRNG generated AES key from TEP Service.
  - PCIE EP WRITE path encryption uses dedicated AES key set (AES_WR)
  - PCIE EP READ response path encryption uses dedicated AES key set (AES_RD)
  - PIP implemented 2x AES key set for AES_WR and 2xAES key set for AES_RD. Only one key sets is active at one time, once that key set counter expires the PIP switches to the second key set. The software is notified by interrupt about the key set expiration.

10.6 Multi Processor System

The Multi Processor System (MPS) module is used for OS communication via shared memory.

10.6.1 Overview

The Symmetric Multiprocessors (SMP) are single address space processors. As the two processors operating in parallel share data, they must coordinate when operating on shared data. This coordination is called synchronization and is performed by look variables, also named as semaphores. The shared data resides in external DDR SDRAM. In addition, the MPS provides a SRAM used to exchange pointers via a Mailbox. Therefore, this internal SRAM is named as Mailbox Memory.

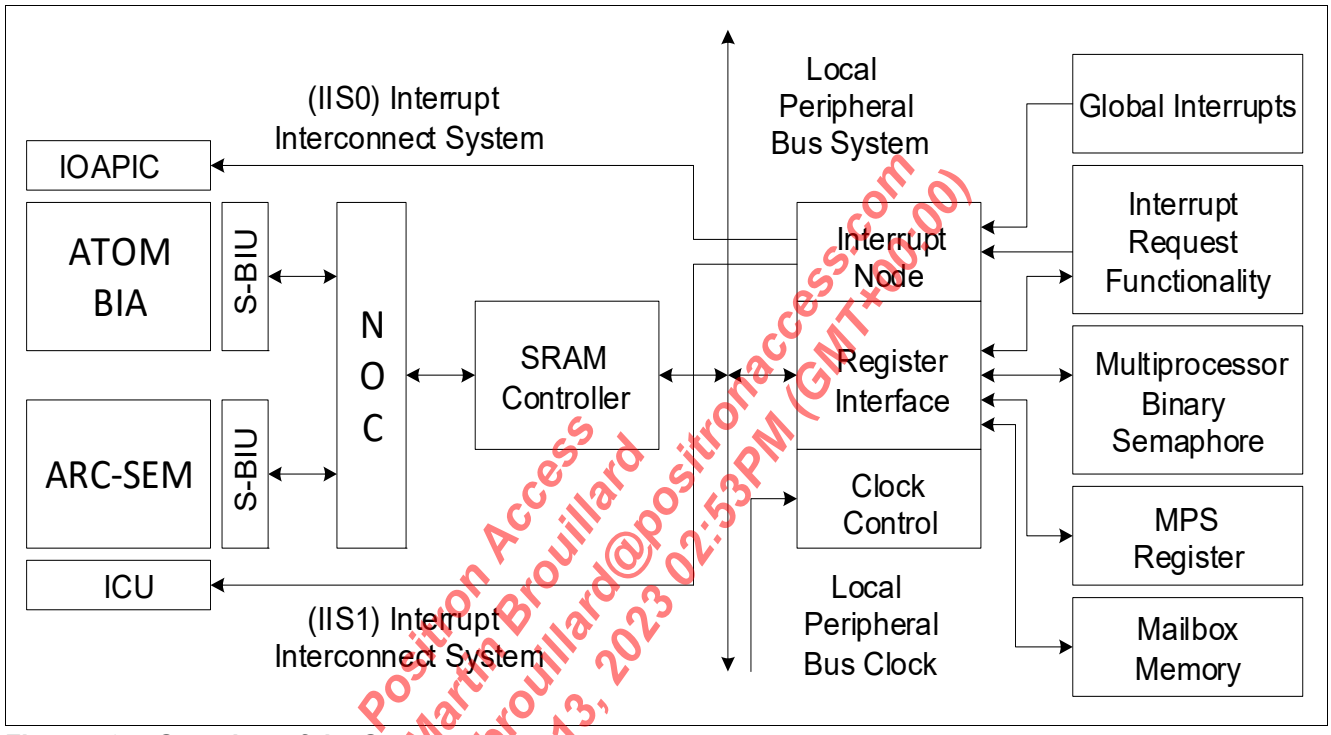


Figure 79 Overview of the System

10.6.2 Features

The MPS provides these features:

- SRAM for Mailbox Memory (1KB)
- Multiprocessor Binary Semaphores
- Interrupt Requests Functionality
- MPS Register

11 Protected Platform Mode

The product support a production enabled Protected Platform Mode (PPM) to prevent illegal access or malicious software to be installed. The PPM description covers these sections:

- **Features** (**Section 11.1**)
- **Functional Description** (**Section 11.2**)

11.1 Features

The ppm supported feature are:

- Post chip production enabled protected platform mode via mode bits programmed in OTP
- EJTAG access protection via protected pin mux
  - Level 1
    EJTAG ES mode fuse bit to allow engineering sample testing and ROM_EJTAG_Reg to allow software after ROM code to re-enable EJTAG for non-secure standard product or jump to level 2
  - Level 2
    The software communicates with SPE [aka Vault-130], after all required authentication and digital sign verification, SPE enables the EJTAG path (SPE Debug [0], SPE Debug [2], SPE Debug [7]).
  - Intel Atom core supervisor mode
    It is possible to enable/disable the Intel Atom Core for the daisy chain regardless of the modes by using the OTP fuse reflected in OEMC register bit 2.
  - PON IP supervisor mode
    The PON IP uses the signal as GPIO input on the module to be evaluated by the PON firmware.

11.2 Functional Description

There are two protection modes:

- EJTAG Access Protection
- UART Console Access Protection

11.2.1 Protected JTAG

The productive chip devices have the default e-fuse setting that JTAG for boot ROM stage is protected (no EJTAG). The ODM/OEM is not able to see the ROM code execution using EJTAG probe, only later parts of the software images could be debugged.

Secure EJTAG Modes

The scheme has these features:

- Level 2 mode
  The software communicate with SPE [aka Vault-130], after all required authentication and digital sign verification, SPE enables the EJTAG path even when OEMC Register Bit 2 is provisioned to '1' [Secure EJTAG enabled]. There are ports used on SPE:
  - Debug Port [0] = TEP Core Secure
  - Debug Port [1] = PON (not valid for URX850)
  - Debug Port [2] = Intel Atom Core
  - Debug Port [5] = TEP Core normal
  - Debug Port [6]= UART Console
  - Debug Port [7] = others Core
- Secure Core Supervisor Mode
  - The Intel Atom Core can be enabled/disabled from the daisy chain regardless of the modes by provisioning of OEMC register bit 2 [ATOM_ISO].

Secure UART Console Mode

The scheme has these features:

- The software communicates with SPE. After all required authentication and digital sign verification, the SPE enables the UART console path even when OEMC register bit 2 is provisioned to 1 (Secure EJTAG/UART enabled). This is the port used on SPE.
  - Debug Port [6]= UART Console
- Selects which UART as console:
  - After secure EJTAG is enabled, the software must program the **Security_TOP register offset 0xD4** to select which UART is being used as console port. By default, it is UART0 as console.
  - The authenticated UART/console port Dbg[6] is used to unlock the selected UART.

12 Power Management

This chapter describes the power management aspects of systems. Clock related power saving features are implemented in the CGU. All hardware power management functionality is coordinated and controlled by system software (firmware) and made available to user via power management API. As general strategy power saving is performed local when possible and central where required by constraints.

The Power Management Unit (PMU), also called Energy Processing Unit (EPU) is described in **Section 12.2**.

The URX851/URX850/MxL25641 integrates temperature sensor to sense silicon junction temperature for system power managements inputs and system thermal management inputs. Refer to **Section 15.3.3 Built-in Temperature Sensor** for sensor accuracy.

12.1 Features

The PMU works on a 40 MHz oscillator clock and its features are:

- Central hardware support for SoC power gating management by EPU
- Up to five power rails defined on SoC level for core voltage supply:
  - Rail0 (VDDD0V8CPU0L2) for Module 0 CPU and L2SRAM/C6RAM
  - Rail1 (VDDD0V8CPU1L2) for Module 1 CPU and L2SRAM/C6RAM (not available for MxL25641)
  - Rail2 (VDDD0V8ADP) for Accelerated Data Path (ADP). No power switchers around ADP power domain integrated, ON/OFF is from external board circuit.
  - Rail3 (VDDD0V8ROC) for the rest of the SoC
  - Rail4 (VDDD0V8DDR) for DDR of running DDR interface higher or equal to 3733 MT per second. The DDR power rail is combined with Rail3 (VDDD0V8ROC) in MxL25641. This is possible as the product support a maximum. DDR speed grade of up to 3200.
- Support battery backup unit mode
- Control of power domain supply voltage
  - Dynamic switching on anf off of the individual Intel Atom core power domain voltages via P-unit (8051) inside the BaseIA subsystem.
  - Automated power domain logical isolation control
  - Dynamic Voltage Scaling (DVS)
- Dynamic Frequency Scaling (DFS) working together with the CGU
- ADP (Accelerated Data Path) blocks, such as GSWIP-O, PPv4, and CQM, can have multiple operating points via DVFS/ DFS to balance performance against power efficiency requirements.
  - Software-triggered ADP DVFS
  - Hardware-autonomous ADP DFS

12.1.1 Power Domains

This section describes the SoC power domains.

Table 129 Power Rails and Domains

IP	Description	Power Domain
CPU_M0_CORE0	BaseIA Module0 Core0	Yes
CPU_M0_CORE1	BaseIA Module0 Core1	Yes
CPU_M0_L2	BaseIA Module0 L2	Yes
CPU_M1_CORE0[1)]	BaseIA Module1 Core0	Yes
CPU_M1_CORE1[1)]	BaseIA Module1 Core1	Yes
CPU_M1_L2[1)]	BaseIA Module1 L2	Yes
Uncore	BaseIA Uncore	Yes
HSIO1	Dual SerDes Super Macro 1	Yes
HSIO2	Dual SerDes Super Macro 2	Yes
HSIO3[1)]	Dual SerDes Super Macro 3	Yes
HSIO4[1)]	Dual SerDes Super Macro 4	Yes
Voice	Voice Subsystem	Yes
RoC	NoC, Security Engines, rest digital of the chip	No
PON[2)]	PON Subsystem	Yes
AON	Always On IPs	No
DDR	DDR Controller	Yes
ADP	Accelerated Data Path, including GSWIP-O, PPV4, BM, QoS, and CQEM	Yes

1) Not available in MxL25641
2) Not available in URX850.

When considering the power against the cost trade-off, a 5-rail (high BOM), and 4-rail (medium BOM) plus other rails configurations are recommended for URX851/URX850. The high BOM is the most recent configuration, which only difference with the medium BOM configuration is an additional DDR voltage rail (0.9 V).

Note: For both configurations, the Power Management Integrated Circuit (PMIC) must be used as power source.

Table 130 URX851/URX850 High BOM Configuration

Rail ID	Power Rail	Power Domain	Scalable
Rail0	V080_CPU_M0	PD_CPU_M0 and PD_L2RAM_M0	Yes
Rail1	V080_CPU_M1	PD_CPU_M1 and PD_L2RAM_M1	Yes
Rail2	V080_ADP	PD_ADP	Yes
Rail3	V080_RoC	PD_Voice	Yes
		PD_SoC	
		PD_PON[1)]	
		PD_HSIO1	
		PD_HSIO2	
		PD_HSIO3	
		PD_HSIO4	
		PD_AON	
Rail4	V090_DDR	PD_DDR	No
Other rails	D1V80		No
	A0V80		
	A1V80		
	A1V10		
	A3V30		

1) Not available in URX850.

When considering the power against the cost trade-off, a 3-rail (maximum power saving BOM) is recommended for MxL25641. It is possible to combine V080_ADP and V080_RoC to save BOM cost by trading off the ability to use DVS on rail2. Combining all three rails (0, 2, and 3) is possible but removes the DVFS ability for the CPU rail, which requires software changes and adds system power.

Table 131 MxL25641 Maximum Power Saving BOM Configuration

Rail ID	Power Rail	Power Domain	Scalable
Rail0	V080_CPU_M0	PD_CPU_M0 and PD_L2RAM_M0	Yes
Rail2	V080_ADP	PD_ADP	Yes
Rail3	V080_RoC	PD_Voice	Yes
		PD_SoC	
		PD_PON	
		PD_HSIO1	
		PD_HSIO2	
		PD_AON	
Other rails	D1V80		No
	A0V80		
	A1V80		
	A1V10		
	A3V30		

Table 132 URX851/URX850 Medium BOM Configuration

Rail ID	Power Rail	Power Domain	Scalable
Rail0	V080_CPU_M0	PD_CPU_M0 and PD_L2RAM_M0	Yes
Rail1	V080_CPU_M1	PD_CPU_M1 and PD_L2RAM_M1	Yes
Rail2	V080_ADP	PD_ADP	Yes
Rail3	V080_RoC	PD_Voice	No
		PD_SoC	
		PD_PON[1)]	
		PD_HSIO1	
		PD_HSIO2	
		PD_HSIO3	
		PD_HSIO4	
		PD_AON	
		PD_DDR[2)]	
Other rails	D1V80		No
	A0V80		
	A1V80		
	A1V10		
	A3V30		

1) Not available in URX850.

2) Maximum speed grade in this configuration DDR4-3200/LPDDR4-3200

12.1.2 URX851/URX850 to PMIC Interface

Table 133 describes the SoC-PMIC interface signals for URX851/URX850.

Table 133 URX851/URX850 PMIC Interface Signals

Ball	Signal	No. of Signals	Dir.	Description	Notes
EA54 -I2C0SDA DV52-I2C0SCL	I^2C	2	BiDir	Communication interface between the PMIC and SoC. Both the host CPU and EPU inside the SoC access the PMIC.	Mandatory interface for all types of Power Delivery System (PDS)
EJ37 - IO90_PMICIO5	CPU IRQ	1	PMIC to SoC	Interrupt signal to the host CPU, indicating various PDS events such as Voltage Regulator (VR) failures, VR status, thermal event, power button, and other features.	Advanced feature interface. May not be used for low cost systems.
EB38- IO96_PMICIO0	General PG/ POWERGOOD	1	PMIC to SoC	General Power Good (PG) indication to the SoC (RCU and POR). Indicates that all the regulator's output are stable within +/-% of the target value. The SoC POR also senses the AON/RoC rail. This signal and the AON sensing indicate to the SoC POR that all rails are stable and the internal RESET signal can be released.	This signal is required to be Open-Drain. Mandatory interface for all types of PDS.
EB27- IO97_PMICIO1	Request	1	SoC to PMIC	Signal from the SoC to the PDS following events such as watchdog expiration, thermal events, or power mode transition. This signal triggers the PMIC to shut-down, re-boot, change VR output, or any operation defined in this spec.	Mandatory interface for all types of PDS
EC32- IO98_PMICIO2	CPU0 Status/ EPU IRQ	1	PMIC to SoC	Status signal to the SoC EPU, indicating: 1. CPU0 power settle/reject status following a new Voltage Identification Code (VID) request in a non-EPU IRQ based system. 2. IRQ signal to the EPU indicating settle/reject status following a new VID request in an EPU IRQ based.	Mandatory interface for all types of PDS
EL34- IO102_PMICIO3	CPU1 Status	1	PMIC to SoC	Status signal to the SoC EPU, indicating: 1. CPU1 power settle/reject status following a new VID request in a non-EPU IRQ based system.	
EE35 - IO103_PMICIO4	ADP Status	1	PMIC to SoC	Status signal to the SoC EPU, indicating: 1. ADP power settle/reject status following a new VID request in a non-EPU IRQ based system.	

Table 134 describes the SoC-PMIC interface signals for MxL25641.

Table 134 MxL25641 PMIC Interface Signals

Ball	Signal	No. of Signals	Dir.	Description	Notes
BE5 - I2C0SDA-PMIC BE6 -I2C0SCL-PMIC	I2C	2	BiDir	Communication interface between the PMIC and SoC. Both host CPU and EPU inside SoC access the PMIC.	Mandatory interface for all types of PDS
A1 - PMIC_GPIO5	CPU IRQ	1	PMIC to SoC	Interrupt signal to the host CPU, indicating various PDS events such as VR failures, VR status, thermal event, power button, and other features.	Advanced feature interface; may not used for low cost systems
C6- PMICIO0	General PG/ POWERGOOD	1	PMIC to SoC	General PG indication to the SoC (RCU and POR). Indicates that all the regulator's output are stable within +/-% of the target value. The SoC POR also senses the AON/RoC rail. This signal and the AON sensing indicate to the SoC POR that all rails are stable and the internal RESET signal can be released.	This signal is required to be Open-Drain. Mandatory interface for all types of PDS.
C5- PMICIO1	Request	1	SoC to PMIC	Signal from the SoC to the PDS following events such as watchdog expiration, thermal events, or power mode transition. This signal triggers the PMIC to shut-down, re-boot, change VR output or any operation defined in this spec.	Mandatory interface for all types of PDS
D3-PMICIO2	CPU0 Status/ EPU IRQ	1	PMIC to SoC	Status signal to the SoC EPU, indicating: 1. CPU0 power settle/reject status following a new VID request in a non-EPU IRQ based system. 2. IRQ signal to the EPU indicating settle/reject status following a new VID request in an EPU IRQ based.	Mandatory interface for all types of PDS
B6-PMICIO3	Not required[1)]	--	--	--	
A2 - PMICIO4	ADP Status	1	PMIC to SoC	Status signal to the SoC EPU, indicating: 1. ADP power settle/reject status following a new VID request in a non-EPU IRQ based system.	

1) It was the CPU1 status in 4-core products

12.2 EPU

Power management approaches have had its evolution from primitive forms of OS managed power control to micro-controller based form, micro-controller and hardware accelerators, to complete hardware-controlled power management in the recent years to adapt to new use cases, workloads, stringent latency requirements, aggressive power targets prescribed by Code-of-Conduct, Energy-Star, etc.

12.2.1 EPU Overview

For URX851, URX850, and MxL25641, the EPU is a pure hardware-based unit, which provides a flexible and configurable options to realize the needed power management functions.

Figure 80 illustrates the EPU.

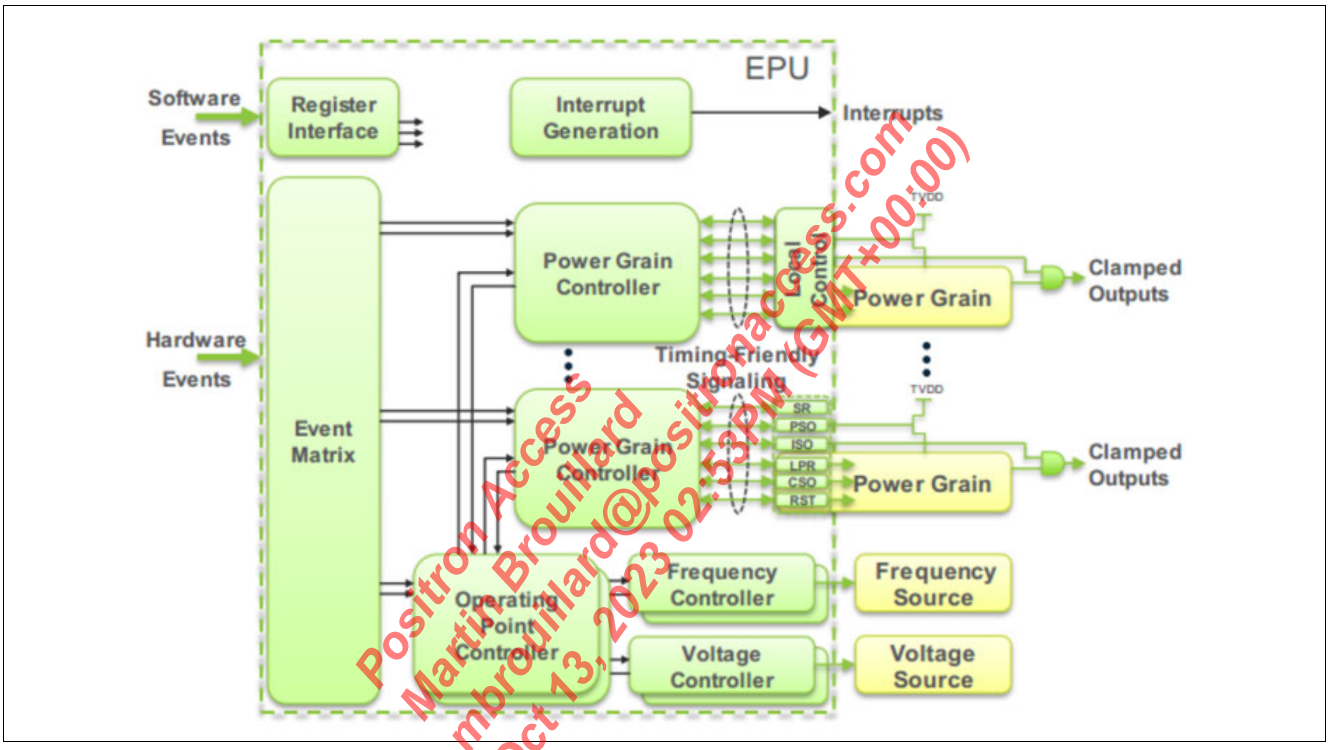


Figure 80 Energy Processing Unit Overview

The EPU contains these blocks:

- The power grain controller or power domain controller manages power control mechanisms for a IP or group of IPs. Power control mechanisms such as clock-gate enable/disable, clamp enable/disable, power-gate enable/disable are examples.
- The Operating Point Controller (OPC) abstracts the state for a collection of grains, where each grain is either a power domain, clock domain, voltage domain or frequency domain. The combined state of the grains is defined as an operating point. The OPC has a voltage controller which talks to the PMIC and a frequency controller that talks to the CGU in the SoC.
- The register interface is the interface through which the software programs the EPU, or issue software events to the EPU.
- The interrupt generation block is used to generate an interrupt to the CPU.

12.2.2 EPU Design in URX851/URX850/MxL25641

The EPU consists of a set of Power Domain Controllers (PDCs) per power domain and a set of OPCs per voltage rail, and then interfaces with various SoC blocks such as temperature sensors, RCU, CGU, and external PMIC via the I^2C interface.

12.2.3 EPU Clusters and Grains

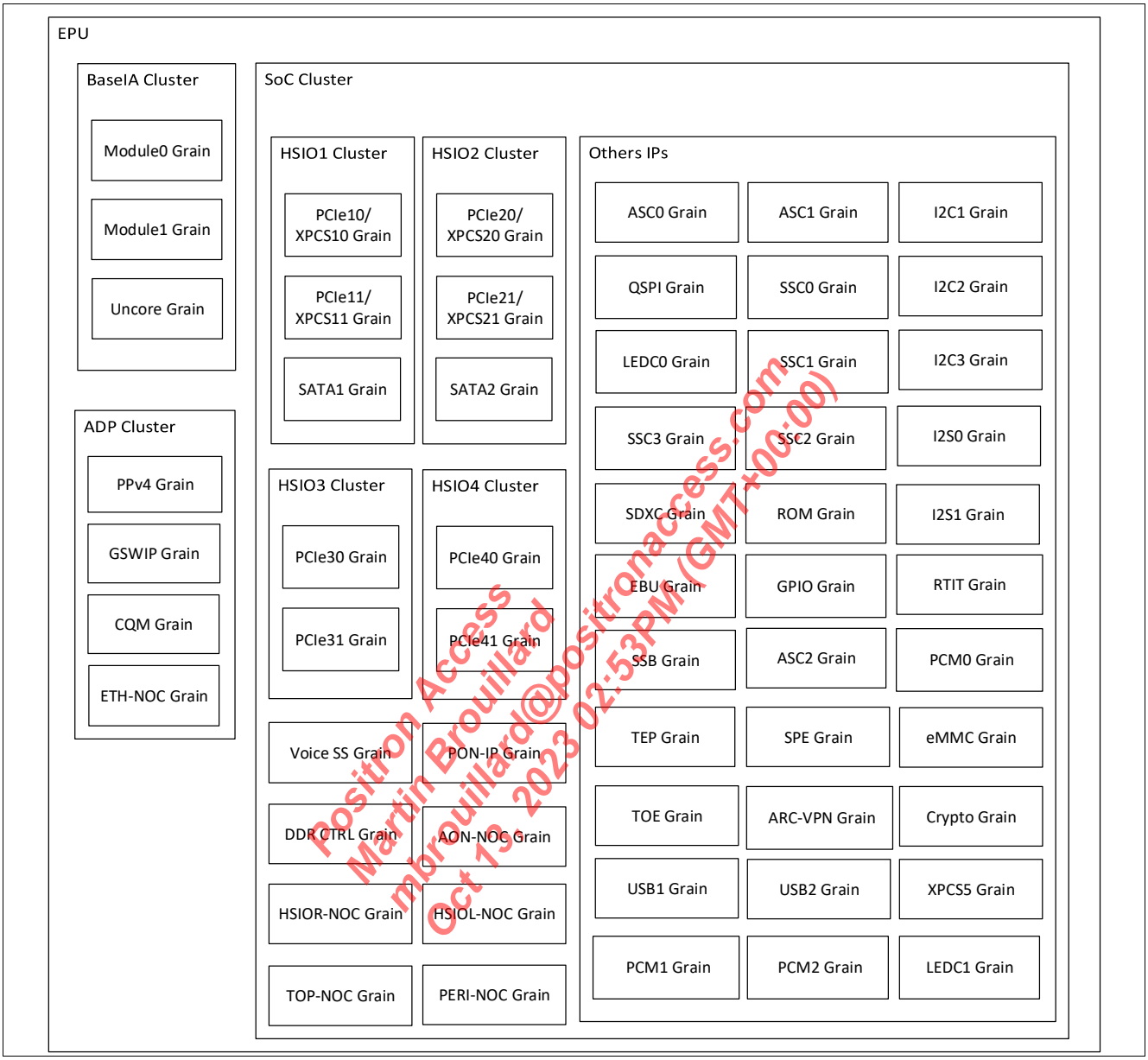


Figure 81 EPU Clusters and Grains

Attention: The PON-IP Grain is not available in URX850.

Attention: These grains are not available in MxL25641:
Module1; PCIe30/31, PCIe40/41; SATA1; SATA2; SATA3; SATA4; SDXC; USB1 (also referred as USB port 0).

It is possible for each grain to be individually power controlled, such as clock gated or power gated, depending on the design. The clusters are constructed in such a way that the grains share clock/voltage sources, allowing for DVFS possibilities, such as BaseIA DVFS or ADP DVFS.

12.2.4 EPU Driver

The EPU driver is a platform specific driver that mediates between Linux/Android PM framework and EPU hardware. **Figure 82** provides a high level overview of the EPU Driver, and how it plugs into the OS framework.

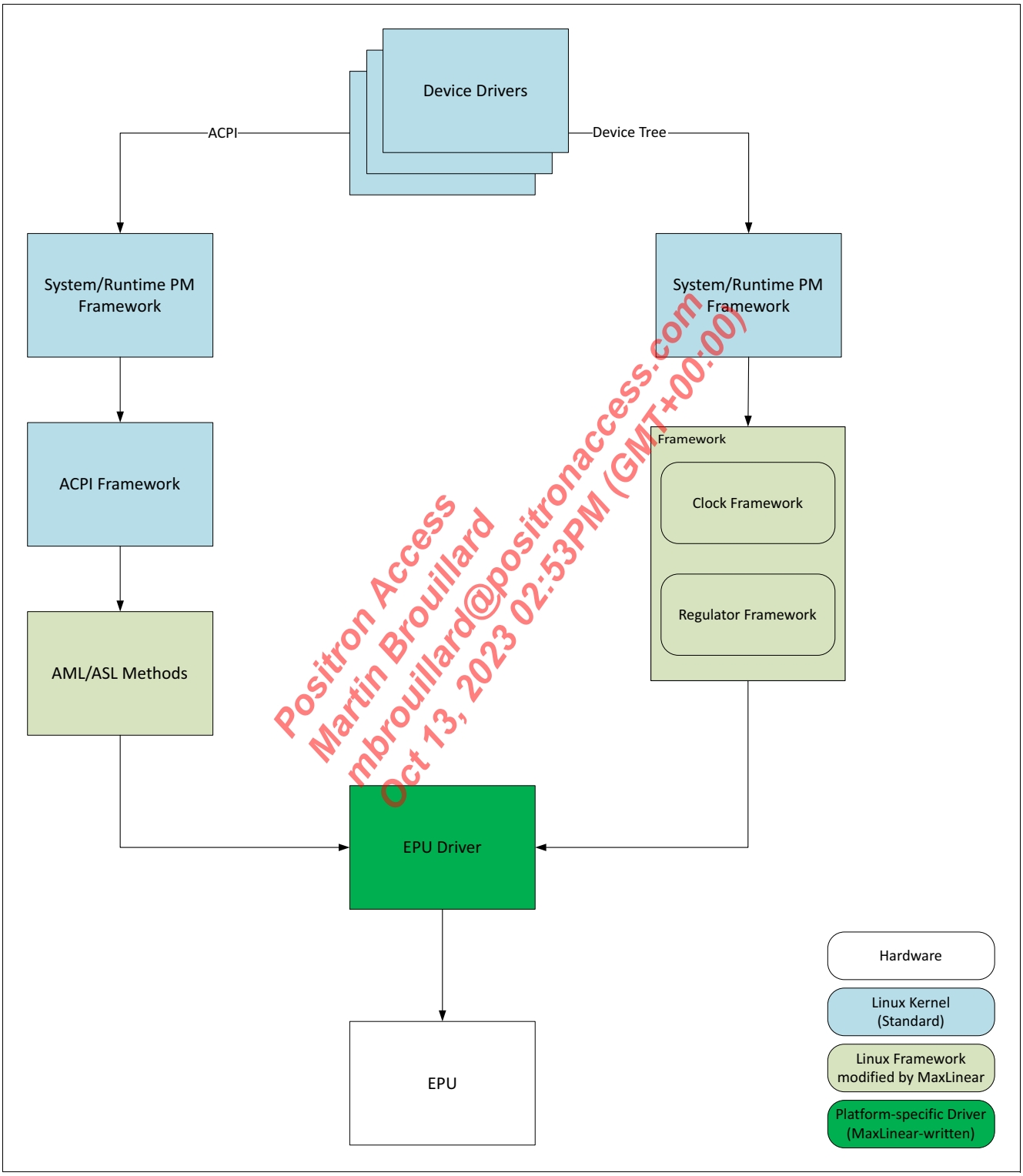


Figure 82 EPU Driver Overview

12.2.4.1 EPU Configuration

Configuring the EPU is one of the primary functions of the EPU driver after cold boot event as part of `epu_init()` routine called at the start. As per the BOM configuration of the part, the PDCs/ OPCs must be configured first, before the registers and SRAM inside the VFT tables and PASEQMEM are programmed. Via the CGU driver interface, the frequency encoding tables must be programmed for each CGU PLL, such that frequency change operations from EPU can go through. These features can be disabed at boot by default. This must be enabled in the EPU. for example ADP DVFS. Finally, the `DEVICE_x_PM_CONTROL` must be programmed to put the various devices in the SoC into appropriate D-state as prescribed by the software. See **Section 12.2.4.5**.

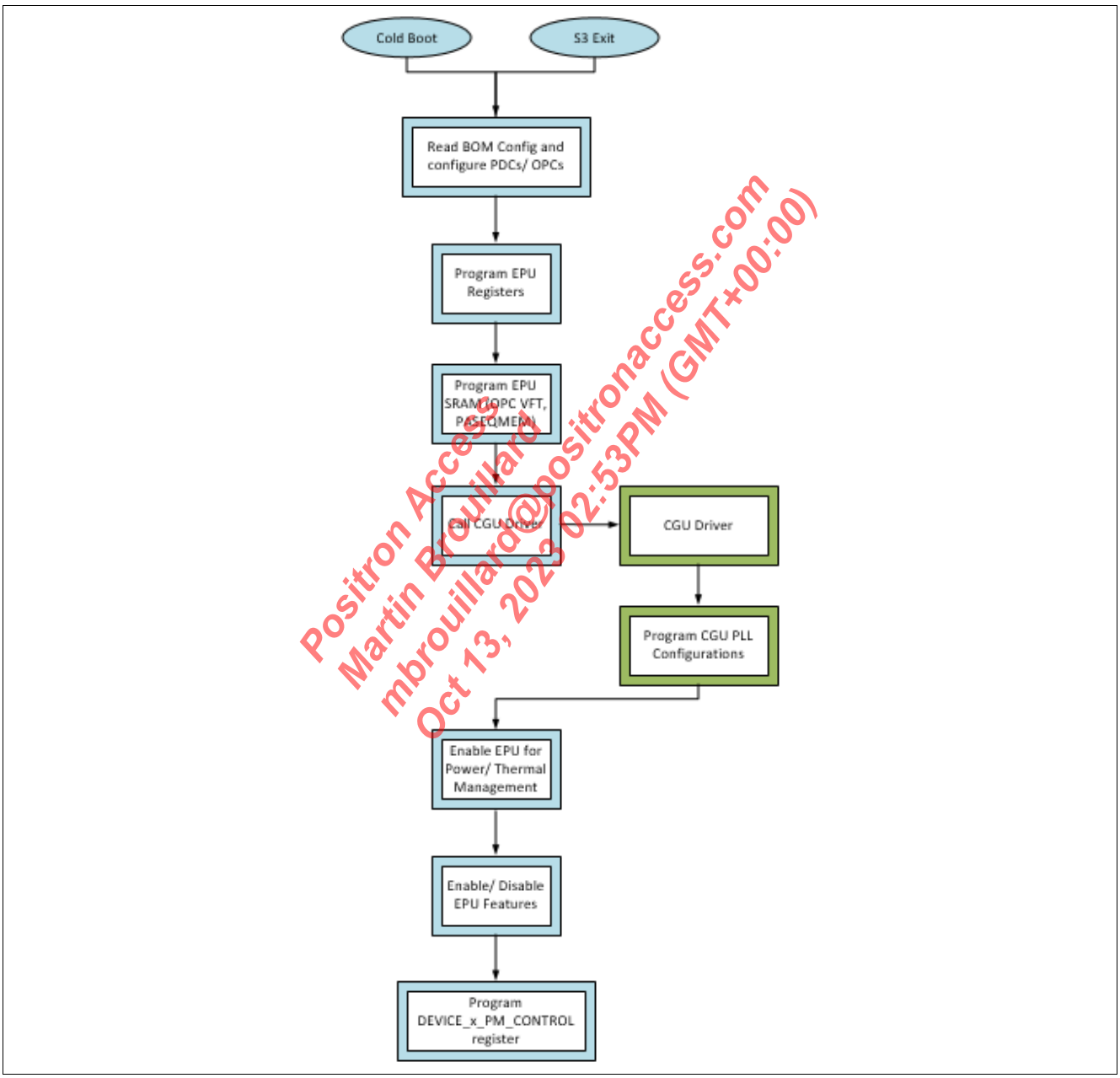


Figure 83 EPU Configuration

Attention: The EPU must be configured very early in the cold boot flow and S3 exit flow.
The software team can decide whether:
a) U-Boot/ FSB must perform this EPU configuration or
b) wait until the OS loads the EPU driver to perform this EPU configuration.

12.2.4.2 Device Initialization

Each of the device drivers using the Linux GenPD PM framework request for attach/detach. This flows through the EPU driver using the callbacks functions `attach_dev()` and `detach_dev()` to translate the software requests to hardware requests by writing to the EPU registers.

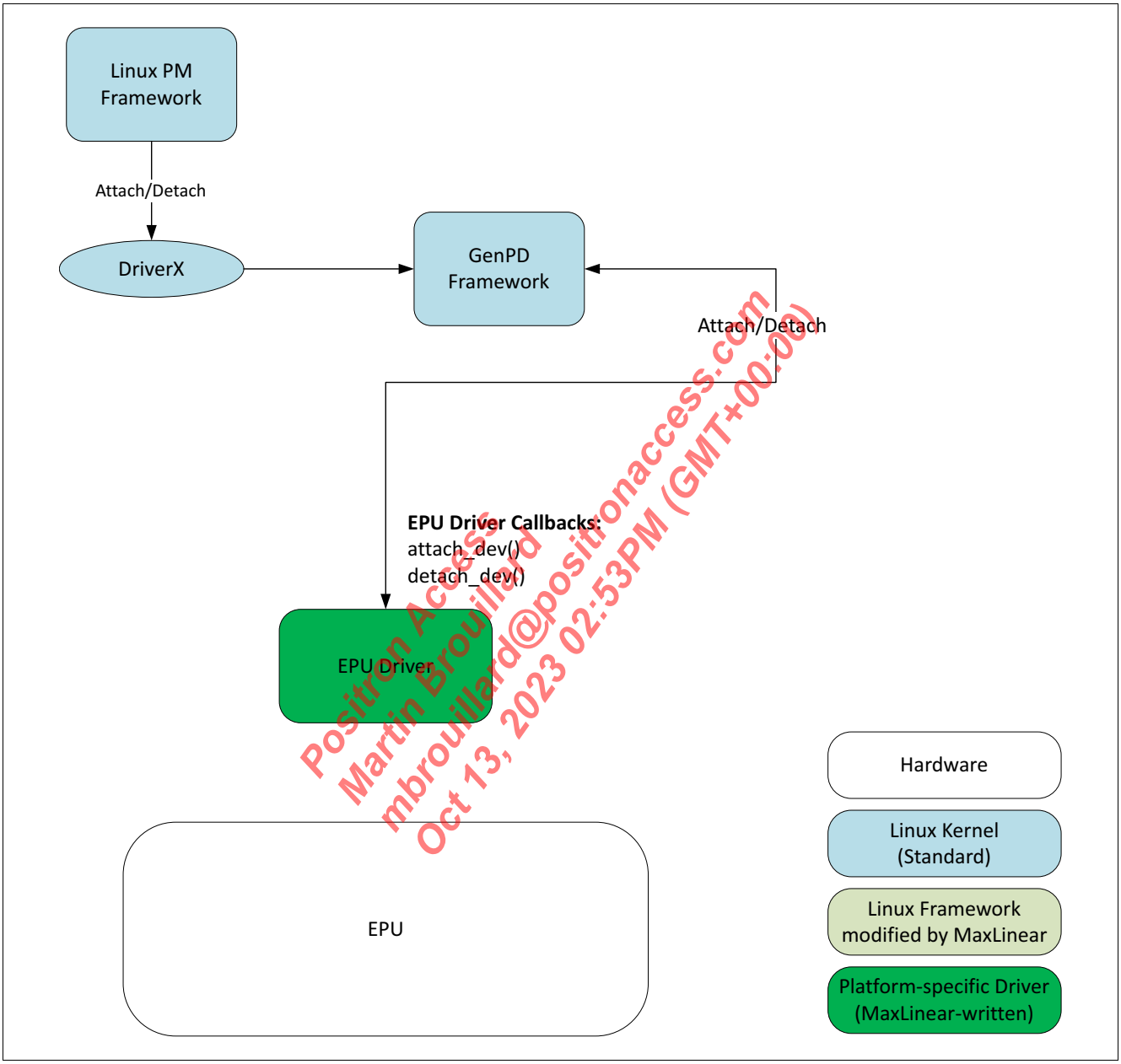


Figure 84 Runtime Suspend/Resume

12.2.4.3 Device Runtime Suspend/ Resume

Each of the device drivers using the Linux GenPD PM framework request for suspend/resume via D-state changes. This flows through the EPU driver using the callbacks functions `power_on()`, `power_off()`, `start_dev()`, and `stop_dev()` to translate the software requests to hardware requests by writing to the EPU registers located in the AON domain. It also reads back and reports the status registers for low power actions.

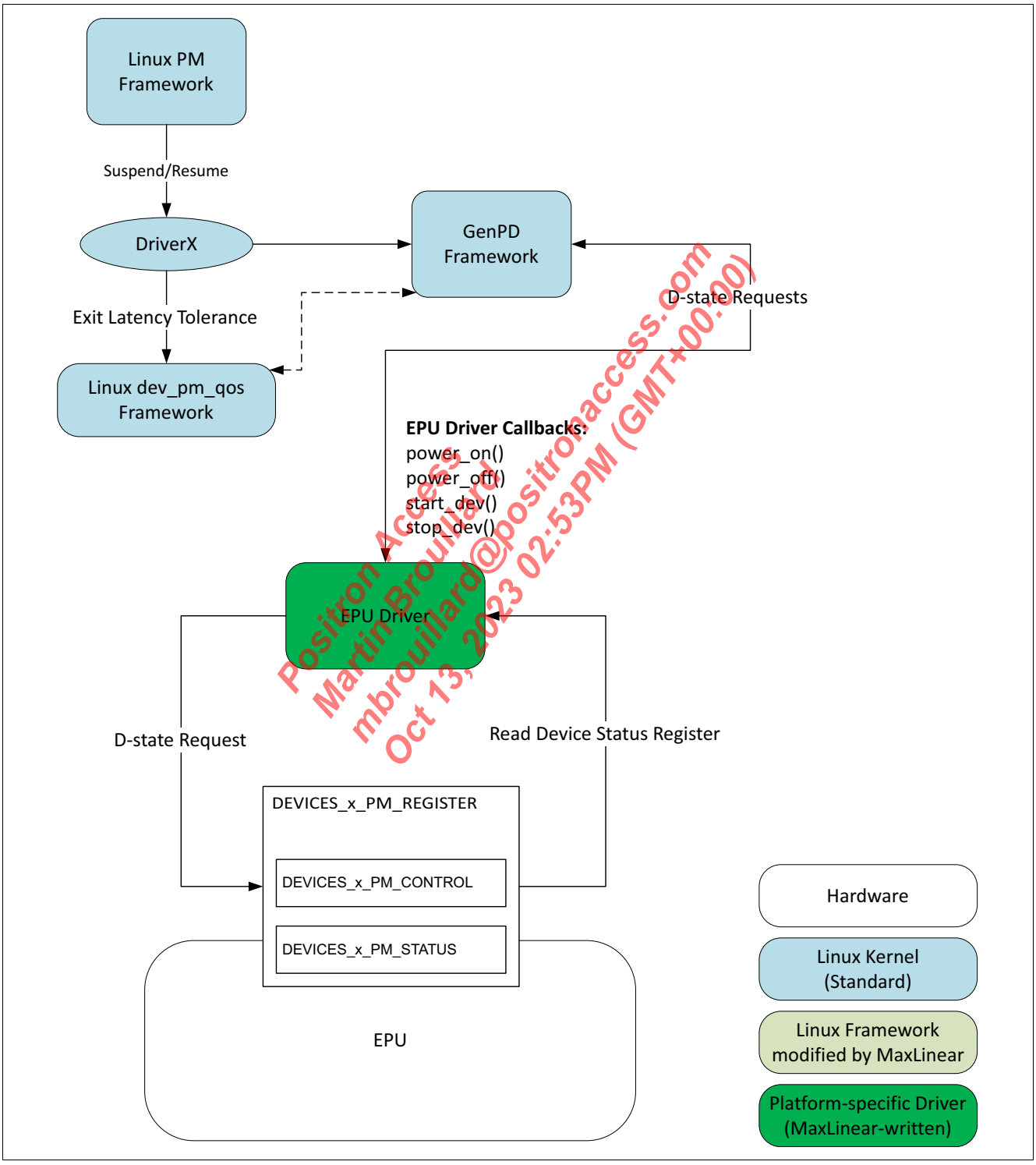


Figure 85 Device Runtime Suspend/Resume

Refer to [5] for details on the callback routines.

12.2.4.4 DVFS

The `setclkrate()` calls to the EPU driver must be translated to a DFS or DVFS request and sent to the EPU. The `getclkrate()` calls to the EPU driver must be forwarded to the CGU driver for obtaining the `clkrate` information, since that functionality is already in the CGU.

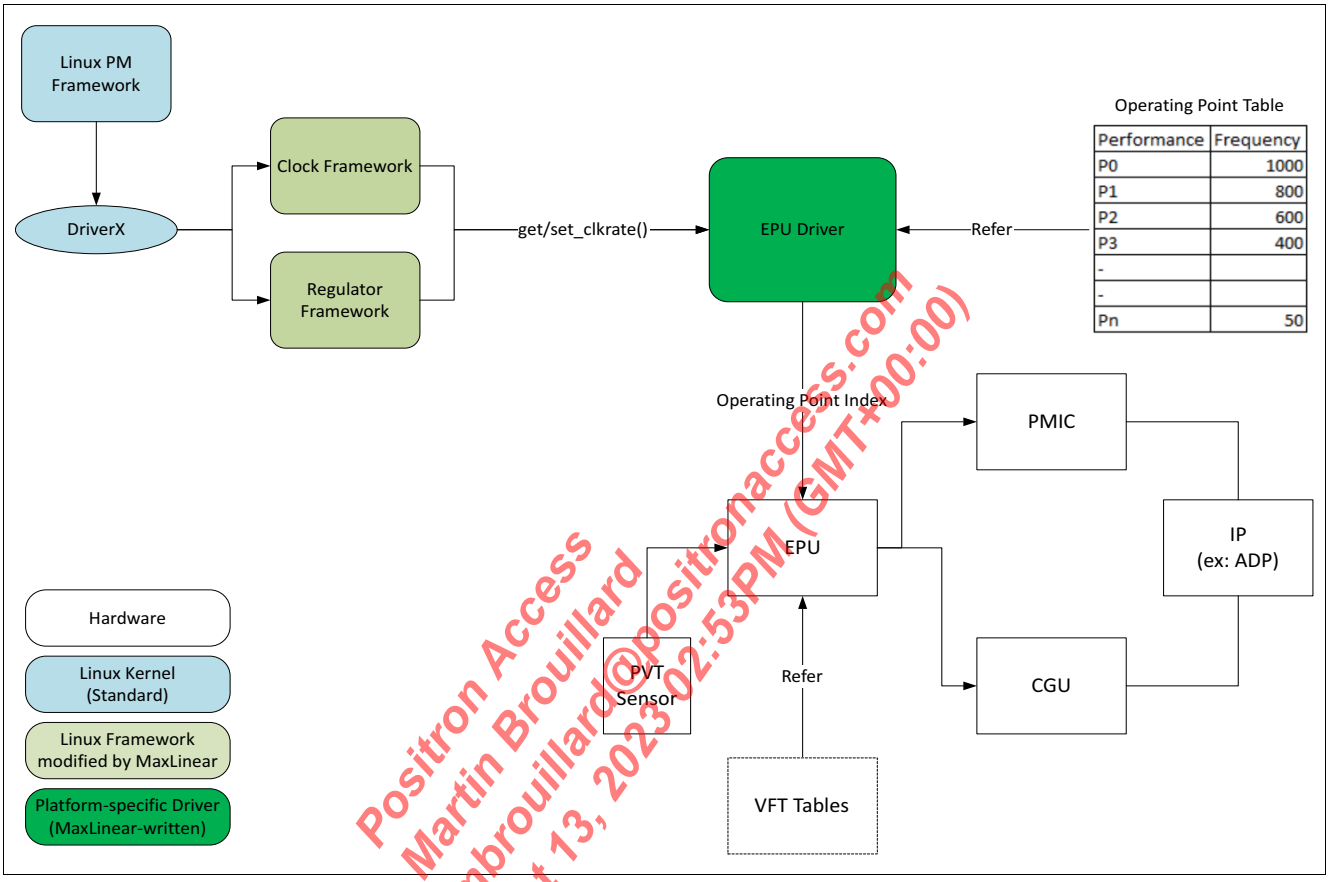


Figure 86 System Blocks involved in DVFS Flow

12.2.4.5 Boot-time Default Devices D-state

Upon cold boot or warm boot, since there is a power cycle for all the IPs in the device, all the devices come to D0 D-state by default, and are ready for action even before the device drivers are loaded. This wastes power.

As per the software teams request, the devices must be in idle state (D1, D3), and only the respective device drivers must bring up relevant devices to D0, as per use case.

For example, during boot under CoC test scenario, only certain devices must be active to run the tests while unused devices stay in idle states saving power. The software EPU driver programs a D-state pattern for all the devices upon boot. According to the pattern, devices go to various D-states to be able to meet the use case requirements and minimize power consumption.

As shown in **Figure 83**, the `DEVICE_x_PM_CONTROL` register must be programmed to put the various devices in the SoC into appropriate D-state. The software remains flexible to update any point of time with a driver update.

12.2.4.6 System State Transition

The system power state transitions, such as S0ix or BBU, are communicated to the EPU driver by the Linux PM framework via programming `SYS_PM_REGISTER`.

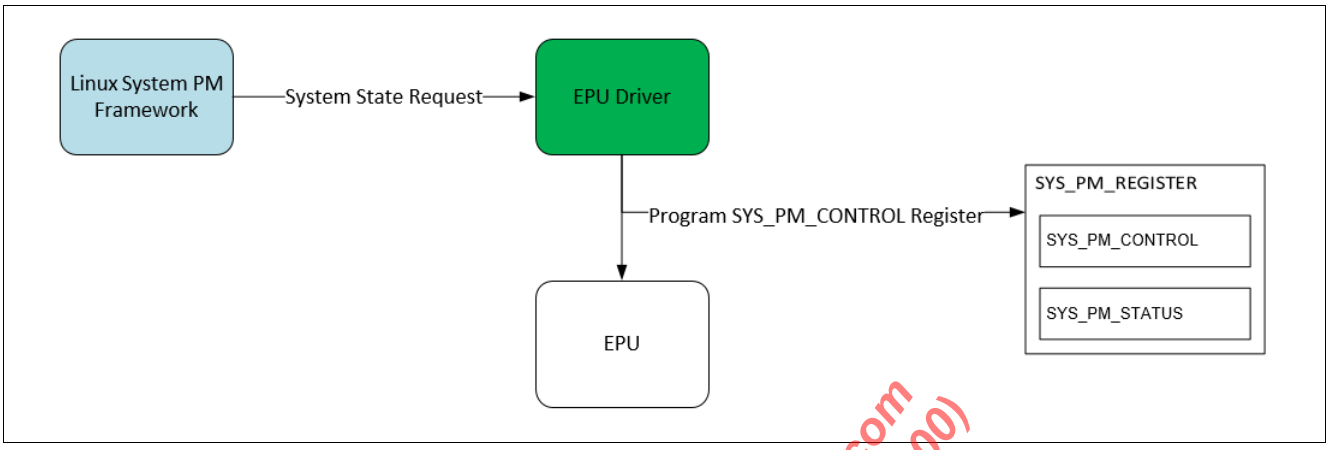


Figure 87 System State Transition

12.2.4.6.1 Suspend-to-Idle and S0ix Support

The EPU driver must perform certain register reads/writes to facilitate Suspend-to-Idle and S0ix entry flow. Figure 88 describes this support.

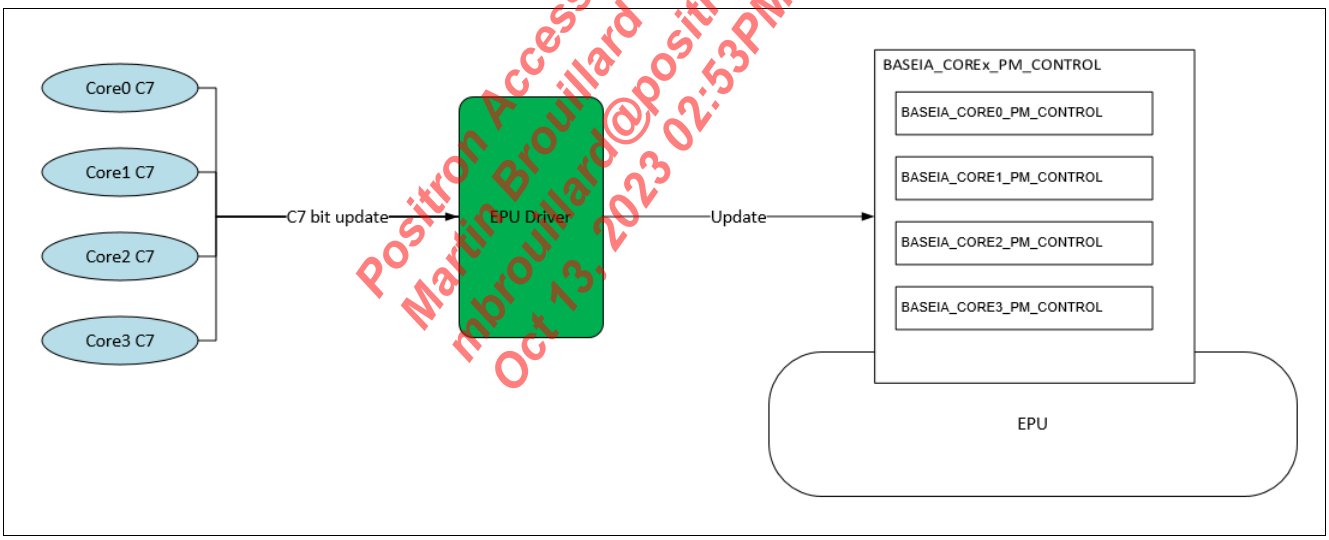


Figure 88 Suspend-to-Idle/S0ix Support

The C7 bits in the `BASEIA_COREx_PM_CONTROL` registers are defined as software trigger BIA S0, where:

- 0 means no BIA S0ix, no system S0ix.
- 1 means software enabling of BIA S0ix, possible system S0ix.

13 Package Outlines and Thermal Parameters for URX851/URX850

The product is assembled in PG-FCBGA-837-C-1A package which complies with regulations requiring lead free material. It is ROHS 6/6 compliant.

Table 135 Thermal Resistance Package Parameter (JEDEC)

Item	Description/Value
Package Type	PG-FCBGA-837-C-1A
Thermal Resistance Junction 1 to Ambient	$R_{th,\ J1A}$ = 22.37 K/W
Thermal Resistance Junction 2 to Ambient	$R_{th,\ J2A}$ = 50.58 K/W
Thermal Resistance Junction 1 to Case - Top	$R_{th,\ J1Ctop}$ = 0.025 K/W
Thermal Resistance Junction 2 to Case - Top	$R_{th,\ J2Ctop}$ = 0.076 K/W
Thermal Resistance Junction 1 to Case - Bottom	$R_{th,\ J1Cbot}$ = 13.16 K/W
Thermal Resistance Junction 2 to Case - Bottom	$R_{th,\ J2Cbot}$ = 28.06 K/W
Thermal Characterization Junction 1 to Case - Top	Ψ_{J1Ctop} = 0.0232 K/W
Thermal Characterization Junction 2 to Case - Top	Ψ_{J2Ctop} = 0.077 K/W
Thermal Characterization Junction 1 to Case - Bottom	Ψ_{J1Cbot} = 9.762 K/W
Thermal Characterization Junction 2 to Case - Bottom	Ψ_{J2Cbot} = 21.51 K/W

Note:

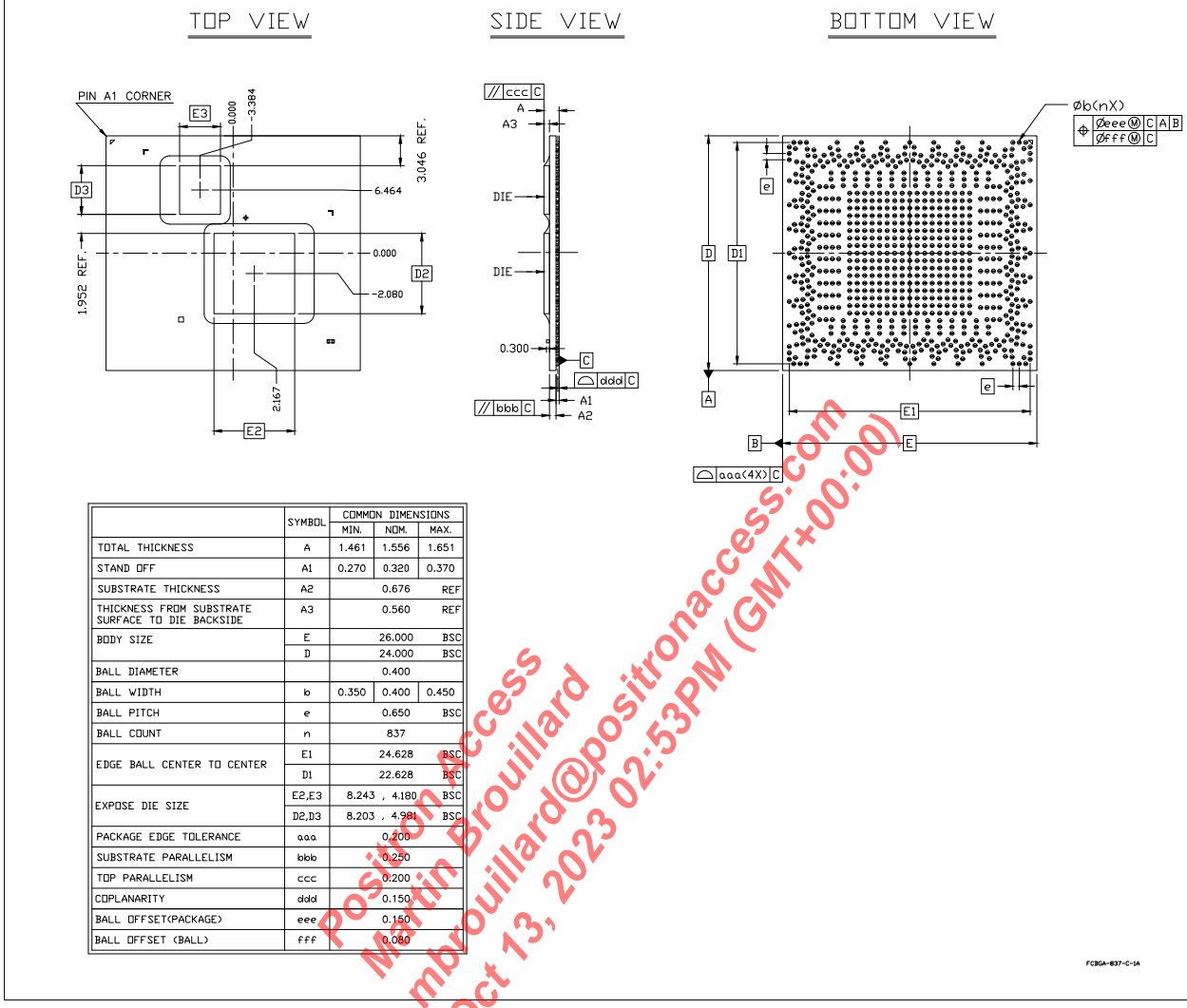
1. *Junction 1 is the bigger die and Junction 2 the smaller die.*
2. *The above values are based on JEDEC standard thermal simulation condition.*
3. *JESD51-9 PCB condition (4 layer PCB of 101.5x114.5 mm, thickness 1.6 mm) is used for Ψ_{JT} and Ψ_{JS}.*
4. *28x28x6 mm heatsink is used for Ψ_{JS} from the vendor Aavid and part number is 106326-27.*

Table 136 5R Thermal Resistance Model for Thermal System Simulation

Item	Description/Value
Package Type	PG-FCBGA-837-C-1A
Thermal Resistance Junction 1 to Case - Top	$R_{th,\ J1Ctop}$ = 0.04 K/W
Thermal Resistance Junction 2 to Case - Top	$R_{th,\ J2Ctop}$ = 0.12 K/W
Thermal Resistance Junction 1 to Case - Bottom	$R_{th,\ J1Cbot}$ = 4.2 K/W
Thermal Resistance Junction 2 to Case - Bottom	$R_{th,\ J2Cbot}$ = 10.4 K/W
Thermal Resistance Junction 1 to Junction 2	$R_{th,\ J1J2}$ = 22.7 K/W

Note:

1. *Junction 1 is the bigger die and Junction 2 the smaller die.*
2. *Refer to **[1]** for the detailed system design simulation setup.*



	SYMBOL	COMMON DIMENSIONS		
		MIN.	NOM.	MAX.
TOTAL THICKNESS	A	1.461	1.556	1.651
STAND OFF	A1	0.270	0.320	0.370
SUBSTRATE THICKNESS	A2		0.676	REF
THICKNESS FROM SUBSTRATE SURFACE TO DIE BACKSIDE	A3		0.560	REF
BODY SIZE	E		26.000	BSC
	D		24.000	BSC
BALL DIAMETER			0.400	
BALL WIDTH	b	0.350	0.400	0.450
BALL PITCH	e		0.650	BSC
BALL COUNT	n		837	
EDGE BALL CENTER TO CENTER	E1		24.628	BSC
	D1		22.628	BSC
EXPOSE DIE SIZE	E2,E3		8.243 , 4.180	BSC
	D2,D3		8.203 , 4.981	BSC
PACKAGE EDGE TOLERANCE	aaa		0.100	
SUBSTRATE PARALLELISM	bbb		0.250	
TOP PARALLELISM	ccc		0.200	
COPLANARITY	ddd		0.150	
BALL OFFSET(PACKAGE)	eee		0.150	
BALL OFFSET (BALL)	fff		0.080	

Figure 89 PG-FCBGA-837-C-1A **(Plastic Green Low-Profile Fine Ball Grid Array) – Bottom View**

Note: The package is sourced from multiple vendors.

Package description, package handling, PCB and board assembly information, ball names and XY coordinates are available on request.

13.1 Ordering Information

This section describes the product and packaging information of the URX851 and URX850 devices.

Table 137 Product and Package Naming

Product Name	Ordering Code	Package	SKU Details
AnyWAN™ SoC URX851	URX851B1BE	PG-FCBGA-837-C-1A	Table 138
AnyWAN™ SoC URX850	URX850B1BE	PG-FCBGA-837-C-1A	Table 139

Table 138 URX851 SKU Details

Item	Description
Product Name	AnyWAN™ SoC URX851
Model Number	**URX851B1BE**
MMID	99LS51
Spec Code	SLS51
Chip Code	URX851
Package	PG-FCBGA-837-C-1A
PON/XFI	PON/XFI
HSIO PCIe 3.0 Instance	4
HSIO Combo (PCIe 3.0/ XFI/ SATA) Instance	4
HSIO Combo (PCIe 3.0/ XFI) Instance	0
2.5GE PHY Interfaces	4
Intel Atom Cores	4
Clock / GHz per Core	2.0 GHz
Packet Processor Performance Class, Mpps/Gbps	30/35
USB 3.2 Gen 2x1 Type-A and Type-C	2
IPsec Performance Class	10 Gbps
Extended Temperature Range[1)]	Yes
eMMC Interface	5.1
8/16 Bit ONFI NAND	Yes
ECC for DDR	Yes
SD Card Interface	Yes
DDR Encryption	Yes

1) -20°C to 85°C, to be confirmed.

Table 139 URX850 SKU Details

Item	Description
Product Name	AnyWAN™ SoC URX850
Model Number	**URX850B1BE**
MMID	99LS50
Spec Code	SLS50
Chip Code	URX850
Package	PG-FCBGA-837-C-1A
PON/XFI	XFI
HSIO PCIe 3.0 Instance	4
HSIO Combo (PCIe 3/ XFI/ SATA) Instance	4
HSIO Combo (PCIe 3/ XFI) Instance	0
2.5GE PHY Interfaces	4
Intel Atom Cores	4
Clock / GHz per Core	2.0 GHz
Packet Processor Performance Class, Mpps/Gbps	30 Mpps/35 Gbps
USB 3.1 Gen 2 Type-A and Type-C	2
IPsec Performance Class	10 Gbps
Extended Temperature Range[1)]	Yes
eMMC Interface	5.1
ONFI NAND	8/16 bit
ECC for DDR	Yes
SD Card Interface	Yes
DDR Encryption	Yes

1) -20°C to 85°C, to be confirmed.

14 Package Outlines and Thermal Parameters for MxL25641

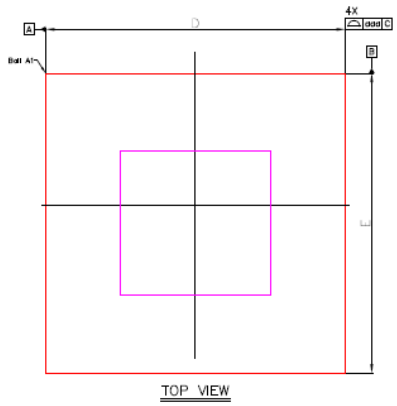
The product is assembled in PG-FCBGA-577 package which complies with regulations requiring lead free material. It is ROHS 6/6 compliant.

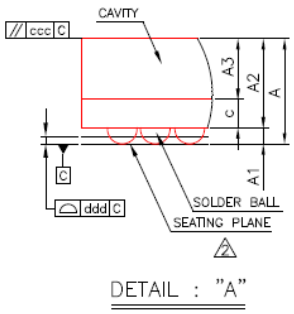
Table 140 Thermal Resistance Package Parameter (JEDEC)

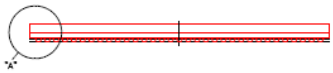
Item	Description/Value
Package Type	PG-FCBGA-577
Thermal Resistance Junction to Ambient	$R_{th,\ JA}$ = 9.49 K/W
Thermal Resistance Junction to Case - Top	$R_{th,\ JCtop}$ = 0.07 K/W
Thermal Resistance Junction to Case - Bottom	$R_{th,\ JCbot}$ = 2.97 K/W
Thermal Characterization Junction to Case - Top	Ψ_{JCtop} = 0.01 K/W
Thermal Characterization Junction to Case - Bottom	Ψ_{JCbot} = 2.99 K/W

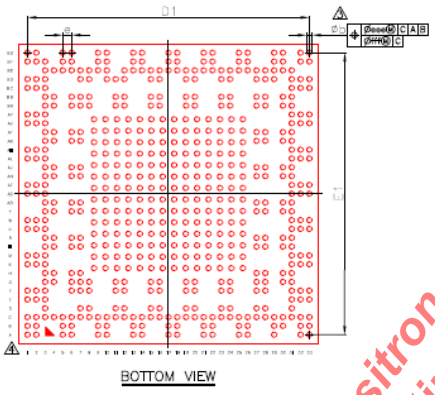
Notes

1. *These values are based on JEDEC standard thermal simulation condition.*
2. *JESD51-9 PCB condition (4 layer PCB of 101.5x114.5 mm, thickness 1.6 mm) is used for Ψ_{JT} and Ψ_{JS}.*
3. *28x28x6 mm heatsink is used for Ψ_{JS} from the vendor AAVID and part number is 106326-27.*









Symbol	Dimension in mm			Dimension in Inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	0,921	0,992	1,063	0,036	0,039	0,042
A1	0.160	0.210	0.260	0.006	0.008	0.010
A2	0.732	0.782	0.832	0.029	0.031	0.033
A3	0,420	0,450	0,480	0,017	0,018	0,019
c	0.292	0.332	0.372	0.011	0.013	0.015
D	16.900	17.000	17.100	0.665	0.669	0.673
E	16.900	17.000	17.100	0.665	0.669	0.673
D1	—	16.000	—	—	0.630	—
E1	—	16.000	—	—	0.630	—
e	—	0.500	—	—	0.020	—
b	0,260	0,310	0,360	0,010	0,012	0,014
aaa		0.150			0.006	
ccc		0.150			0.006	
ddd		0.130			0.005	
eee		0.150			0.006	
fff		0.050			0.002	
MD/ME		33/33				

NOTE:
1. CONTROLLING DIMENSION : MILLIMETER.
2. PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
3. DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM C.
4. THE PATTERN OF PIN 1 FIDUCIAL IS FOR REFERENCE ONLY.
5. BALL PLACEMENT USE 0.30 mm SOLDER BALL. BGA PAD SOLDER MASK OPENING = 0.275 mm.

Figure 90 PG-FCBGA-577 (Plastic Green Low-Profile Fine Ball Grid Array) – Bottom View

Note: The package is sourced from multiple vendors.

The package description, package handling, PCB and board assembly information, ball names and XY coordinates are available on request.

14.1 Ordering Information

This section describes the product and packaging information of MxL25641.

Table 141 Product and Package Naming

Product Name	Ordering Code	Package	SKU Details
AnyWAN™ SoC MxL25641	MxL25641-AV-T	PG-FCBGA-577	Table 142

Table 142 MxL25641 SKU Details

Item	Description
Product Name	AnyWAN™ SoC MxL25641
Model Number	MxL25641-AV-T
MMID	--
Spec Code	--
Chip Code	MxL25641
Package	PG-FCBGA-577
PON/XFI	PON/XFI
HSIO PCIe 3.0 Instance	4
HSIO Combo (PCIe 3.0/ XFI) Instance	1
HSIO Combo (PCIe 3.0/ XFI) Instance	1
XFI Interfaces	1
Intel Atom Cores	2
Clock / GHz per Core	1.7 GHz
Packet Processor Performance Class, Mpps/Gbps	15 Mpps/30 Gbps
USB 3.2 Gen 2x1 Type-A	1
IPsec Performance Class	5 Gbps
Extended Temperature Range	No
eMMC Interface	5.1
ONFI NAND	8-bit
ECC for DDR	No
SD Card Interface	No
DDR Encryption	No

15 Electrical Characteristics of URX851/URX850/MxL25641

The electrical characteristics describe the interface timings, power supply ranges, interface driving strength and ESD robustness.

15.1 Absolute Maximum Ratings

Table 143 lists the absolute maximum ratings. The nominal values of voltages are:

V_{DDP0} = V_{DDPCIE} =V_{DDPLL0} = 1.8 V

V_{DDDDR4} = 1.2 V (supply domain of DDR4)

V_{DDLP4} = 1.1 V

V_{DDC0} = 0.8 V

V_{DDETH0} = 0.95 V

V_{DD3V3} = 3.3 V

Table 143 Absolute Maximum Ratings (Chip Pad Level)

Parameter	Symbol	Values			Unit	Note/ Test Condition
		Min.	Typ.	Max.		
Ambient Temperature under Bias	T_A	0	–	70	°C	–
Storage Temperature	T_{stg}	-55	–	125	°C	–
Junction Temperature	T_{jABS}	0	–	125	°C	Commercial range T ambient is 0 to 70°C
Junction Temperature	T_{jABS}	–40	–	125	°C	Industrial range T ambient is -40 to 85°C
Moisture Level 3 Temperature	T_{ML3}	–	–	260	°C	According to IPS J-STD 020
IC Supply Voltage (Pads, Digital)	V_{DDP}	V_{DDP0} - 0.3	V_{DDP0}	V_{DDP0} + 0.5	V	–
IC Supply Voltage (Core, Digital)	V_{DDC}	V_{DDC0} - 0.3	V_{DDC0}	V_{DDC0} + 0.30	V	–
IC Supply Voltage (ETH, Low Voltage)	V_{DDETH}	V_{DDC0} - 0.3	V_{DDC0}	V_{DDC0} + 0.30	V	–
IC Supply Voltage (PCIE Analog)	V_{DDpcie}	V_{DDpcie} - 0.3	V_{DDpcie}	V_{DDpcie} + 0.3	V	–
IC Supply Voltage (PLLs)	V_{DDPLL}	V_{DDPLL0} - 0.3	V_{DDPLL0}	V_{DDPLL0} + 0.3	V	–
IC Supply Voltage (DDR4)	V_{DDDDR}	V_{DDDDR4} - 0.3	V_{DDDDR4}	V_{DDDDR4} + 0.3	V	–
IC Supply Voltage (LPDDR4)	V_{DDDDR}	V_{DDLP4} - 0.3	V_{DDLP4}	V_{DDLP4} + 0.3	V	–
Voltage on Any Other Pin with Respect to Ground	V_{max}	-0.3	–	V_{DDP0} + 0.5	V	Unless specified otherwise
ESD Robustness HBM: 1.5 kΩ, 100 pF	$V_{ESD,HBM}$	–	–	1000	V	According to ANSI/ ESDA/JEDEC JS-001
ESD Robustness CDM	$V_{ESD,CDM}$	–	–	250	V	According to JESD22-C101

Attention: Stresses above the maximum values listed in this table may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Maximum ratings are absolute ratings; exceeding only one of these values may cause irreversible damage to the integrated circuit.

15.2 Operating Range

Table 144 defines the maximum values of voltages and temperature which may be applied to guarantee proper operation. These values do specify the supply voltage at package pin. It considers IR drop of package and substrate trace to ensure required voltage at pad level.

$V_{DDP0} = V_{DDPCIE} = V_{DDPLL0} = 1.8$ V

$V_{DDDDR4} = 1.2$ V (supply domain of DDR4)

$V_{DDDLP4} = 1.1$ V

$V_{DDC0} = 0.8$ V

$V_{DDETH0} = 0.95$ V

$V_{DDANA0} = 0.80$ V

$V_{DDDDR3} = 1.35$ V

$V_{DDP3} = 3.3$ V

Table 144 Operating Range (Package Pin)

Parameter	Symbol	Values			Unit	Note / Summary of the Voltage Range
		Min.	Typ.	Max.		
Ambient Temperature	T_A	0	–	70	°C	–
Junction Temperature -Operation	T_{jO}	0 (-20 for extended range)	–	110	°C	The system thermal solution must be designed to accommodate Thermal Design Power (TDP) within this range. Operation above the maximum limit for extended periods may adversely affect long-term reliability of the device.
Ambient Temperature Extended	T_{A_ext}	-20	–	85	°C	The system must be preheated to -20°C for boot up.
Supply Voltage Digital Pads for 1.8 V	V_{DDP}	1.62	V_{DDP0}	1.98	V	1.8 V ±10 %[1]
Supply Voltage Digital Pads for 3.3 V	V_{DDP3}	2.97	3.3	3.63	V	3.3 V ±10 %[1]
Supply Voltage Digital Core	V_{DDC}	0.72	V_{DDC0}	0.88	V	0.8 V ±10 %[1][2]
Supply Voltage ETH Low Voltages	V_{DDETH}	0.85	V_{DDETH0}	1.00	V	0.95 V -10 %/+5%[1]
Supply Voltage PCIe PHY Analog 0.80 V	V_{DDANA}	0.76	V_{DDANA0}	0.88	V	0.8 V -8%/+10 %[1]
Supply Voltage DDR4	V_{DDDDR4}	1.14	V_{DDDDR4}	1.26	V	1.20 V ±5 %[1]
Supply Voltage LPDDR4	V_{DDDLP4}	1.06	V_{DDLP4}	1.16	V	1.10 V ±5 %[1]
Supply Voltage PLLs	V_{DDPLL}	1.71	V_{DDPLL0}	1.98	V	1.8 V -5%/+10 %[1]
Digital Input Voltages	V_{ID}	-0.4	–	3.63	V	Overdrive to 3.3 V IO level
Ground	V_{SSP}	0	–	0	V	–

1) Voltage ripple must be less than 30 mV.

2) For CPU rail, the voltage is defined by the VFT table and may exceed this range.

Attention: Operations above the maximum values listed here for extended periods may adversely affect long-term reliability of the device.

Power Up Sequence for URX851/URX850/MxL25641

There is no defined voltage rail power up sequence except that the ROC rail must be ramped up before the analog 0.8 V supply. MaxLinear recommends copying the reference board design's power circuitry.

15.3 DC Characteristics

This section includes the DC characteristics of the I/O interfaces, as well as the power consumption of defined system use cases.

Table 145 DC Characteristics

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
3.3 V I/O Region						
High Level Input Voltage	V_{IH}	2.0 V	–	V_{DDP} +0.3	V	High level input voltage
Low Level Input Voltage	V_{IL}	– 0.3	–	0.8 V	V	-
High Level Output voltage	V_{OH}	V_{DDP}-0.40	–	–	V	I_{OH} = 2/4/8/12 mA
Low Level Output voltage	V_{OL}	–	–	0.40	V	I_{OL} = 2/4/8/12 mA
1.8 V I/O Region						
High Level Input Voltage	V_{IH}	0.65* V_{DDP}	–	V_{DDP} +0.3	V	High level input voltage
Low Level Input Voltage	V_{IL}	– 0.3	–	V_{DDP} 0.35*	V	V_{OUT} <= $V_{OL\ (max)}$
High Level Output Voltage	V_{OH}	V_{DDP}-0.45	–	–	V	I_{OH} = 2/4/8/12 mA
Low Level Output voltage	V_{OL}	–	–	0.4	V	I_{OL} = 2/4/8/12 mA
1.2 V I/O Region (Nominal/DDR4)						
I/O Supply Voltage DDR4 at 1.2 V	V_{DDR4}	1.14	1.2	1.26	V	1)
DC High Voltage	$V_{IH(DC)}$	0.5* V_{DDR}	–	1.1* V_{DDR}	V	–
DC Low Voltage	$V_{IL(DC)}$	-0.3	–	0.5* V_{DDR}	V	–
1.1 V I/O Region (LPDDR4)						
I/O Supply Voltage LPDDR4 at 1.1 V	V_{LPDDR4}	1.06	1.1	1.16	V	2)
DC High Voltage	$V_{IH(DC)}$	0.65* V_{LPDDR}	–	0.2* V_{LPDDR}	V	–
DC Low Voltage	$V_{IL(DC)}$	-0.2	–	0.35* V_{LPDDR}	V	–

1) Refer to the DDR4 SDRAM device data sheet about the minimum supply voltage.
2) Refer to the LPDDR4 SDRAM device data sheet about the minimum supply voltage.

Table 146 Typical System Power Supply Current for URX851/URX850

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Measurement Condition: Bi-directional 1.25 Gbps from XFI #4 (with external 10G PHY connected) to each of the 2.5GE PHY ports, from XFI #3 (with external 10G PHY connected) to two PCIe 3.0 ports with an external Wi-Fi chip connected 0.5 Gbps to each PCIe port bi-directional traffic VOIP: phone2 call phone1 and off hook; CPU running with 20% load and power saving states are on; LAN cable 30 meters	I_{IO} [1]	–	65	–	mA	V_{DD} = 3.3 V, T_J = 85 °C. System power rail connects to these pins **VDDD3V3GPIO**, **VDDA3v3_USB2.1**, **VDDA3v3_USB2.0**, **VDD3V3PAD**, **VDDP1V8_SDIO**.
	$I_{Core\ cpu0}$ [2]	–	500	–	mA	V_{DDC} = 0.8 V, T_J = 85 °C. System power rail connects to these pins **VDDD0V8CPU0L2**.
	$I_{Core\ cpu1}$ [2]	–	500	–	mA	V_{DDC} = 0.8 V, T_J = 85 °C. System power rail connects to these pins **VDDD0V8CPU1L2**.
	$I_{Core\ adp}$ [2]	–	1500	–	mA	V_{DDc} = 0.8 V, T_J = 85 °C. System power rail connects to these pins **VDDD0V8ADP**.
	$I_{Core\ roc}$ [2]	–	1700	–	mA	V_{DDC} = 0.8 V, T_J = 85 °C. System power rail connects to these pins **VDDD0V8ROC**, **VDDD0V8CPUPLLPOST**, **VDDD0V8CPUPLLREF**, **VDDD0V8LCPLLPOST**, **VDDD0V8LCPLLREF**.
	$I_{Core\ ddr}$ [2]	–	810	–	mA	V_{DDC} = 0.80 V, T_J = 85 °C. System power rail connects to these pins **VDDD0V8DDR**.
	$I_{Core\ Analog1}$ [1]	–	510	–	mA	V_{DDA1} = 0.80 V, T_J = 85 °C. System power rail connects to these pins **VDDA0V8VCEMMC**, **VA0V8_USB1**, **VA0V8_USB0**, **VDDA0V8CML**, **VA_0V8_PCIE10_11**, **VA_0V8_PCIE20_21**, **VA_0V8_PCIE30_31**, **VA_0V8_PCIE40_41**, **VA_0V8_PON**, **VA_0V8_XFI5**.

Table 146 Typical System Power Supply Current for URX851/URX850 (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Measurement Condition: Bi-directional 1.25 Gbps from XFI #4 (with external 10G PHY connected) to each of the 2.5GE PHY ports, from XFI #3 (with external 10G PHY connected) to two PCIe 3.0 ports with an external Wi-Fi chip connected 0.5 Gbps to each PCIe port bi-directional traffic VOIP: phone2 call phone1 and off hook; CPU with 20% load and power saving states are on; LAN cable 30 meters	$I_{Core\ Analog2}$ [1]	–	440	–	mA	V_{DDA2} = 1.80 V, T_J = 85 °C. System power rail connects to these pins **VDDA1V8EMMCQ**, **VPH_1V8_PCIE10_11**, **VPH_1V8_PCIE20_21**, **VPH_1V8_PCIE30_31**, **VPH_1V8_PCIE40_41**, **UVPHA1V8**, **VDDA1V8CML**, **VPH_1V8_USB0**, **VPH_1V8_USB1**, **VDDA1V8LCPLL**, **VDDA1V8CPUPLL**, **VDDA1V8ROPLL**, **VDDA1V8DDRPLL**, **VDDA1V8PORXO**, **DVDD1V8_JTAG**, **VPH_1V8_PON**, **VPH_1V8_XFI5**, **VDDA1V8OTPVT**, **VDDA1V8FUSE**.
	I_{D1V2} [1]	–	141	–	mA	V_{D1v2} = 1.1 V, T_J = 85 °C. System power rail connects to these pins **VDDA1V1DDR**, **VDDA0V6DDR**.
	I_{Eth_Hi} [1]	–	306	–	mA	V_{Eth_Hi} = 3.3 V, T_J = 85 °C. System power rail connects to these pins **VDDHA3V3**.
	I_{Eth_Lo} [2]	–	2660	–	mA	V_{Eth_Lo} = 0.95 V, T_J = 85 °C. System power rail connects to these pins **VDDLA0V9**, **VDD0V9CORE**.

1) Consider a 30% increase on top of the typical current value as maximum current for power regulator design. Refer to the Power Management Application Note [5] and Hardware Design Guide [1] for power and thermal design.

2) Consider a 35% increase on top of the typical current value as maximum current for power regulator design. Refer to the Power Management Application Note [5] and Hardware Design Guide [1] for power and thermal design.

Table 147 Extra High CPU Performance System Power Supply Current for URX851/URX850

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Measurement Condition: Bi-directional 1.25 Gbps from XFI #4 (with external 10G PHY connected) to each of the 2.5GE PHY ports, from XFI #3 (with external 10G PHY connected) to two PCIe 3.0 ports with an external Wi-Fi chip connected 0.5 Gbps to each PCIe port bi-directional traffic, data copying from a SSD connected to one USB port to a SSD connected to the second USB port (100% CPU load) VOIP: phone2 call phone1 and off hook; CPU running with full load at 2 GHz; LAN cable 30 meters	I_{IO} [1]	–	68	–	mA	V_{DD} = 3.3 V, T_J = 100 °C. System power rail connects to these pins **VDDD3V3GPIO**, **VDDA3v3_USB2.1**, **VDDA3v3_USB2.0**, **VDD3V3PAD**, **VDDP1V8_SDIO**.
	$I_{Core\ cpu0}$ [2]	–	2720	–	mA	V_{DDC} = 0.8 V, T_J = 100 °C. System power rail connects to these pins **VDDD0V8CPU0L2**.
	$I_{Core\ cpu1}$ [2]	–	2720	–	mA	V_{DDC} = 0.8 V, T_J = 100 °C. System power rail connects to these pins **VDDD0V8CPU1L2**.
	$I_{Core\ adp}$ [2]	–	2080	–	mA	V_{DDc} = 0.8 V, T_J = 100 °C. System power rail connects to these pins **VDDD0V8ADP**.
	$I_{Core\ roc}$ [2]	–	2570	–	mA	V_{DDC} = 0.8 V, T_J = 100 °C. System power rail connects to these pins **VDDD0V8ROC**, **VDDD0V8CPUPLLPOST**, **VDDD0V8CPUPLLREF**, **VDDD0V8LCPLLPOST**, **VDDD0V8LCPLLREF**.
	$I_{Core\ ddr}$ [2]	–	600	–	mA	V_{DDC} = 0.80 V, T_J = 100 °C. System power rail connects to these pins **VDDD0V8DDR**.
	$I_{Core\ Analog1}$ [1]	–	510	–	mA	V_{DDA1} = 0.80 V, T_J = 100 °C. System power rail connects to these pins **VDDA0V8VCEMMC**, **VA0V8_USB1**, **VDDA0V8CML**, **VA_0V8_PCIE10_11**, **VA_0V8_PCIE20_21**, **VA_0V8_PCIE30_31**, **VA_0V8_PCIE40_41**, **VA_0V8_PON**, **VA_0V8_XFI5**.

Table 147 Extra High CPU Performance System Power Supply Current for URX851/URX850 (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Measurement Condition: Bi-directional 1.25 Gbps from XFI #4 (with external 10G PHY connected) to each of the 2.5GE PHY ports, from XFI #3 (with external 10G PHY connected) to two PCIe 3.0 ports with an external Wi-Fi chip connected 0.5 Gbps to each PCIe port bi-directional traffic, data copying from a SSD connected to one USB port to a SSD connected to the second USB port (100% CPU load) VOIP: phone2 call phone1 and off hook; CPU running with full load at 2 GHz; LAN cable 30 meters	$I_{Core\ Analog2}$ [1]	–	440	–	mA	V_{DDA2} = 1.80 V, T_J = 100 °C. System power rail connects to these pins **VDDA1V8EMMCQ**, **VPH_1V8_PCIE10_11**, **VPH_1V8_PCIE20_21**, **VPH_1V8_PCIE30_31**, **VPH_1V8_PCIE40_41**, **UVPHA1V8**, **VDDA1V8CML**, **VPH_1V8_USB0**, **VPH_1V8_USB1**, **VDDA1V8LCPLL**, **VDDA1V8CPUPLL**, **VDDA1V8ROPLL**, **VDDA1V8DDRPLL**, **VDDA1V8PORXO**, **DVDD1V8_JTAG**, **VPH_1V8_PON**, **VPH_1V8_XFI5**, **VDDA1V8OTPVT**, **VDDA1V8FUSE**.
	I_{D1V2} [1]	–	270	–	mA	V_{D1v2} = 1.2 V, T_J = 100 °C. System power rail connects to these pins **VDDA1V1DDR**, **VDDA0V6DDR**.
	I_{Eth_Hi} [1]	–	330	–	mA	V_{Eth_Hi} = 3.3 V, T_J = 100 °C. System power rail connects to these pins **VDDHA3V3**.
	I_{Eth_Lo} [2]	–	2900	–	mA	V_{Eth_Lo} = 0.95 V, T_J = 100 °C. System power rail connects to these pins **VDDLA0V9**, **VDD0V9CORE**.

1) Consider a 30% increase on top of the typical current value as maximum current for power regulator design. Refer to the Power Management Application Note [5] and Hardware Design Guide [1] for power and thermal design.

2) Consider a 35% increase on top of the typical current value as maximum current for power regulator design; refer to the Power Management Application Note [5] and Hardware Design Guide [1] for power and thermal design.

Table 148 Typical System Power Supply Current for MxL25641

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Measurement Condition: Bi-directional 1.25 Gbps from XFI #4 (with external 10G PHY connected) to each of the 2.5GE and 1.0GE PHY ports, from XFI #3 (with external 10G PHY connected) to two PCIe 3.0 ports with an external Wi-Fi chip connected 0.5 Gbps to each PCIe port bi-directional traffic, data copying from a SSD connected to the USB port (100% CPU load) VOIP: phone2 call phone1 and off hook; CPU running with full load at 1.71 GHz; LAN cable 30 meters	I_{IO} [1]	–	66	–	mA	V_{DD} = 3.3 V, T_J = 85 °C. System power rail connects to these pins **VDDD3V3GPIO**, **VDDA3v3_USB2.1**.
	$I_{Core\ cpu0}$ [2]	–	1560	–	mA	V_{DDC} = 0.8 V, T_J = 85 °C. System power rail connects to these pins **VDDD0V8CPU0L2**.
	$I_{Core\ adp}$ [2]	–	1485	–	mA	V_{DDc} = 0.8 V, T_J = 85 °C. System power rail connects to these pins **VDDD0V8ADP**.
	$I_{Core\ roc}$ [2]	–	1898	–	mA	V_{DDC} = 0.8 V, T_J = 85 °C. System power rail connects to these pins **VDDD0V8ROC**, **VDDD0V8CPULCREF**, **VDDD0V8CPULCPOST**.
	$I_{Core\ Analog1}$ [1]	–	430	–	mA	V_{DDA1} = 0.80 V, T_J = 85 °C. System power rail connects to these pins **VDDA1V8OTPPOREMMC**, **VA_0V8_PCIE10_11**, **VA_0V8_PCIE20_21**, **VA_0V8_PON**, **VA_0V8_USB1_XFI5**, **VDDA0V8CML_LEFT**, **VDDA0V8CML_RIGHT**, **VDDA0V8VCEMMC**.
Measurement Condition: Bi-directional 1.25 Gbps from XFI #4 (with external 10G PHY connected) to each of the 2.5GE and 1.0GE PHY ports, from XFI #3 (with external 10G PHY connected) to two PCIe 3.0 ports with an external Wi-Fi chip connected 0.5 Gbps to each PCIe port bi-directional traffic VOIP: phone2 call phone1 and off hook; CPU running with 20% load at 1.71 GHz; LAN cable 30 meters	$I_{Core\ Analog2}$ [1]	–	380	–	mA	V_{DDA2} = 1.80 V, T_J = 85 °C. System power rail connects to these pins **VDDA1V8OTPPOREMMC**, **VDDA1V8DDRPLL**, **VDDA1V8ROLCPLL**, **VDDA1V8CPUPLL**, **DVDD1V8_JTAG**, **VPH_1V8_PCIE10_11**, **VPH_1V8_PCIE20_21**, **VPH_1V8_PON**, **VDDA1V8CML_LEFT**, **VDDA1V8CML_RIGHT**, **VPH_1V8_USB1_XFI5**, **VDDA1V8FUSE**.
	I_{D1V2} [1]	–	140	–	mA	V_{D1v2} = 1.2 V, T_J = 85 °C. System power rail connects to these pins **VDDA1V1DDR**.

1) Consider a 30% increase on top of the typical current value as maximum current for power regulator design. Refer to the Power Management Application Note [5] and Hardware Design Guide [1] for power and thermal design.
2) Consider a 35% increase on top of the typical current value as maximum current for power regulator design. Refer to the Power Management Application Note [5] and Hardware Design Guide [1] for power and thermal design

15.3.1 POR Detection

Table 149 describes the electrical parameters of the under voltage detection.

Table 149 UVD Characteristics for POR Detection

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Detection Threshold	V_{DET}	0.60	0.64	0.68	V	5% accuracy This is related to the ROC power domain.
Delay for Deactivation of Reset	t_{DEACT}	–	82	–	ms	–

15.3.2 USB Overcurrent Detection or PON Dying Gasp

Table 150 describes the electrical parameters of the USB overcurrent detection or PON dying gasp detection.

Table 150 UVD Characteristics for USB Overcurrent/PON Dying Gasp Detection

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Reference Voltage (1.5 V)	V_{REF}	–	1.5	–	V	Internal voltage, no pin
Hysteresis	V_{Hys}	0	79	-	mV	–

Attention: The input voltage to any of the UVD detection pins must not exceed 3.3 V.

15.3.3 Built-in Temperature Sensor

Table 151 describes the electrical parameters of the integrated temperature sensor.

Table 151 Temperature Sensor Characteristics

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Temperature Range	T_{range}	-40		150	°C	
Resolution			12		bits	
Accuracy		-5	–	+5	°C	Without calibration

15.4 AC Characteristics

T_A = 0 to 70 °C, V_{DD3} = 3.3 V ± 0.3 V, V_{SS} = 0 V

Timing measurements are made at 2.0 V for a logical 1 and at 0.8 V for a logical 0.

Figure 91 shows the AC testing input/output waveforms. The load capacitors are according to the specific interface standard, all non-specified interfaces use 30 pF as assumed loading.

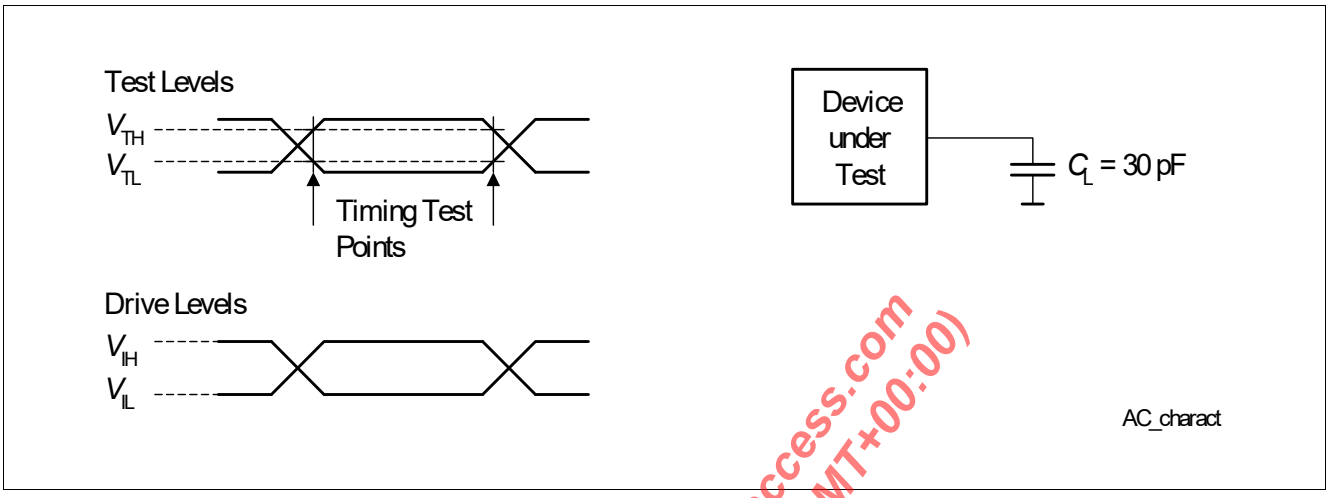


Figure 91 Input/Output Waveform for AC Tests

15.5 Timing Characteristics

The timing characteristics describes the detailed interface timings.

15.5.1 MDIO Interface

Figure 92 shows the timing diagram of the MDIO interface for a clock cycle in the read, write and turnaround mode, respectively. The timing measures are annotated. Table 152 summarizes the defined absolute values.

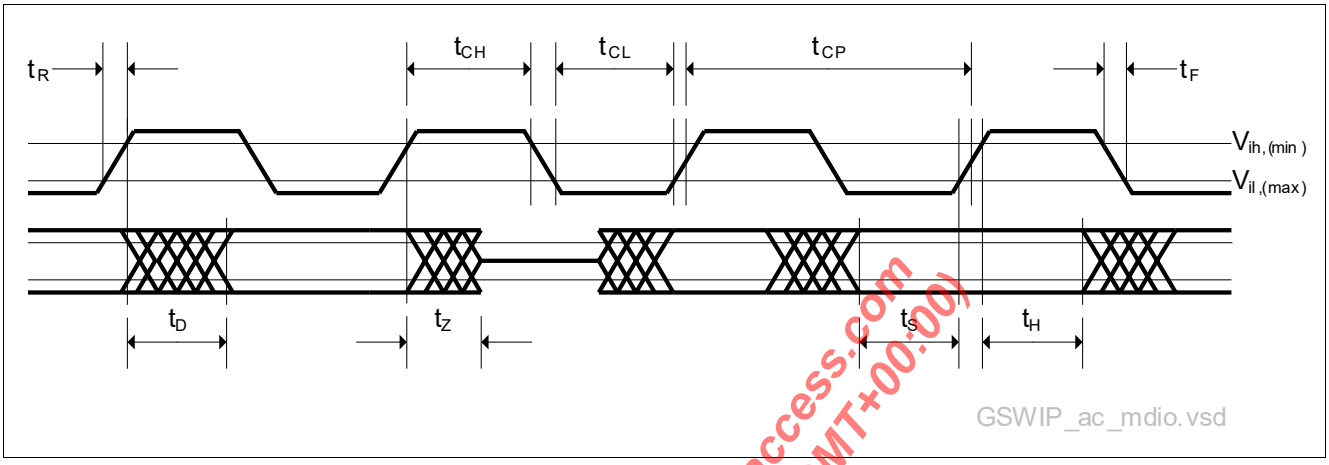


Figure 92 Timing Diagram for the MDIO Interface

Table 152 Timing Characteristics of the MDIO Interface

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
MDC High Time	t_{CH}	10.0	–	–	ns	Given timings are all subject to the MDC at the pin of the URX device.
MDC Low Time	t_{CL}	10.0	–	–	ns	
MDC Clock Period	t_{CP}	40.0	400.0	–	ns	
MDC Clock Frequency[1)]	$1/t_{CP}$	–	2.5	25.0	MHz	
MDC Rise Time	t_R	–	–	5.0	ns	
MDC Fall Time	t_F	–	–	5.0	ns	
MDIO Input Setup Time Subject to - MDC	t_S	8.0	–	–	ns	MAC receive
MDIO Input Hold Time Subject to - MDC	t_H	0.0	–	–	ns	MAC receive
MDIO Output Setup Time Subject to - MDC	t_S	14.0	–	–	ns	MAC transmit
MDIO Output Hold Time Subject to - MDC	t_H	14.0	–	–	ns	MAC transmit
Standard						
MDIO Output Delay Subject to - MDC	t_D	0.0	–	300.0	ns	PHY transmit
MDIO Output Setup Time Subject to - MDC	t_S	10.0	–	–	ns	MAC transmit
MDIO Output Hold Time Subject to - MDC	t_H	10.0	–	–	ns	MAC transmit

1) MDC clock supports range of frequencies, up to 25 MHz. Default/typical frequency is 2.5 MHz.

15.5.2 NAND Flash Interface Timing

This section describes the supported NAND flash timing, in terms of chip select pulse mode.

Only specific NAND flash supports chip select latch mode. The latch mode doesn't require special timing other than what is stated in this section. But it requires the chip select to be always latched to low during the whole NAND page access.

The EBU interface supports the NAND chip select latch mode, however, in this mode, the sharing EBU interface between the NAND device and other devices are mutually exclusive, which means once the NAND access in this mode, the system must prevent simultaneous other EBU device access.

It is achievable by masking interrupt request from other non-NAND EBU devices during this NAND access period.

15.5.2.1 NAND Read Access Timing

Table 153 lists the NAND read access timing.

Table 153 Timing for NAND Data Access Mode

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Chip Select Setup Time Before $\overline{WE}$ Falling Edge	t_{cs}	0			ns	Output load capacitance of 50 pF
CLE Setup Time Before $\overline{WE}$ Falling Edge	t_{cls}	0			ns	
CLE Hold Time After $\overline{WE}$ Rising Edge	t_{clh}	10			ns	
ALE Setup Time Before $\overline{WE}$ Falling Edge	t_{als}	0			ns	
ALE Hold Time After $\overline{WE}$ Rising Edge	t_{alh}	10			ns	
CMD/Addr/Data Output Delay from $\overline{WE}$ Falling Edge	t_{oD}	0		t_{WP} - 15	ns	15 ns is the minimum input setup time.
$\overline{RD}$ Delay to $\overline{READY}$ Rising Edge	t_{RR}	20			ns	
$\overline{RD}$ Low to Data Ready	t_{REA}			20	ns	
$\overline{RD}$ High to Data Hold	t_{OH}	15			ns	
Cycle Time [FPI = 200 MHz]	t_{clk}	-	5.0	-	ns	Cycle time is derived from the programmed FPI Bus clock.
Read: $\overline{RD}$ Evaluation Delay After ALE Falling Edge	t_{AR2}	50			ns	
Read Recovery Cycle	t_{RC}	50			ns	Programmable
Write Enable Pulse Time	t_{WP}	15			ns	
Write Enable Hold Time	t_{WH}	35			ns	Programmable
Write Recovery Cycle	t_{WC}	50			ns	Programmable

Figure 93 and **Figure 94** show the complete timing including the programmable phases.

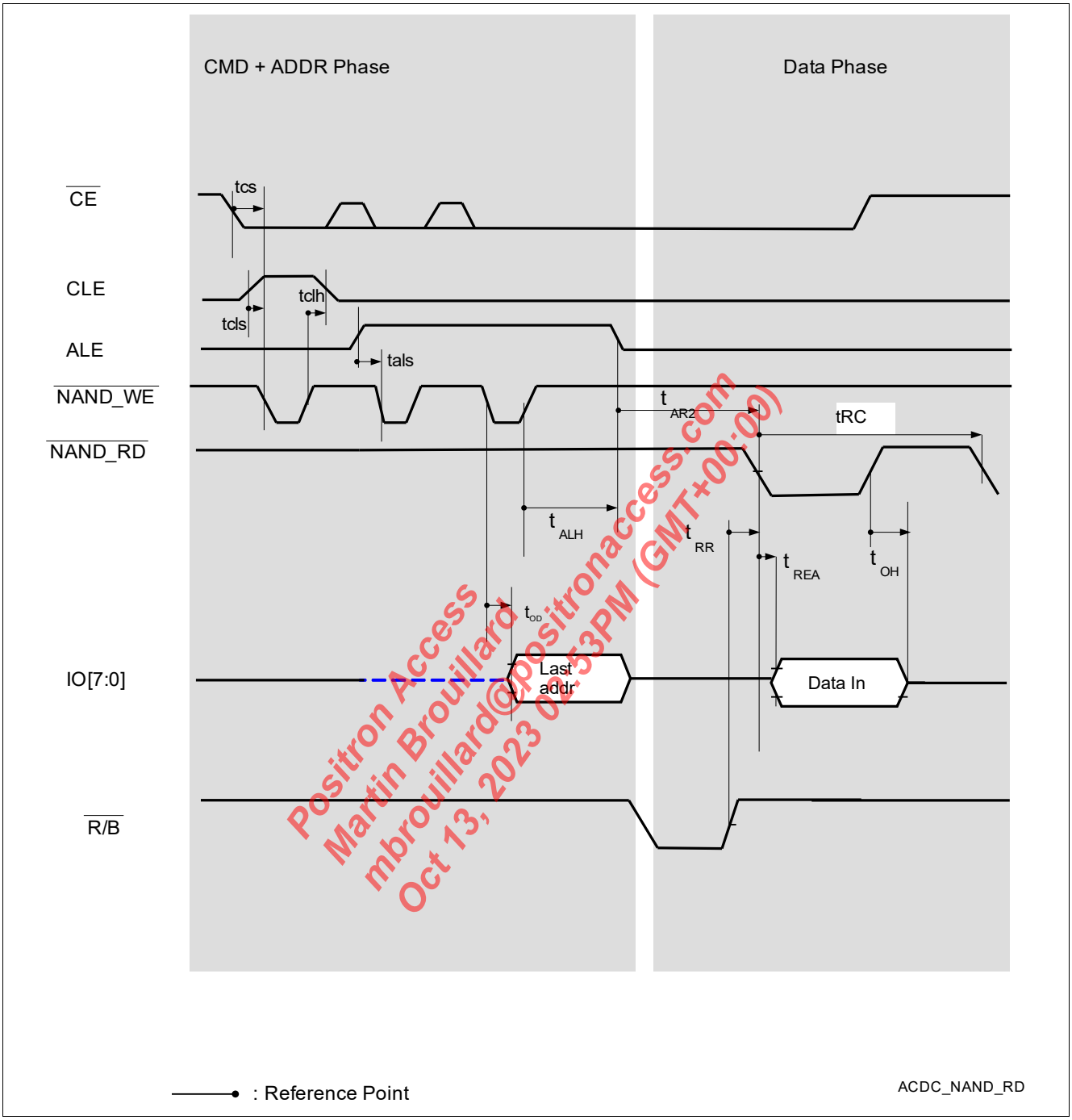


Figure 93 Read Cycle NAND Single Read - Pulse Mode

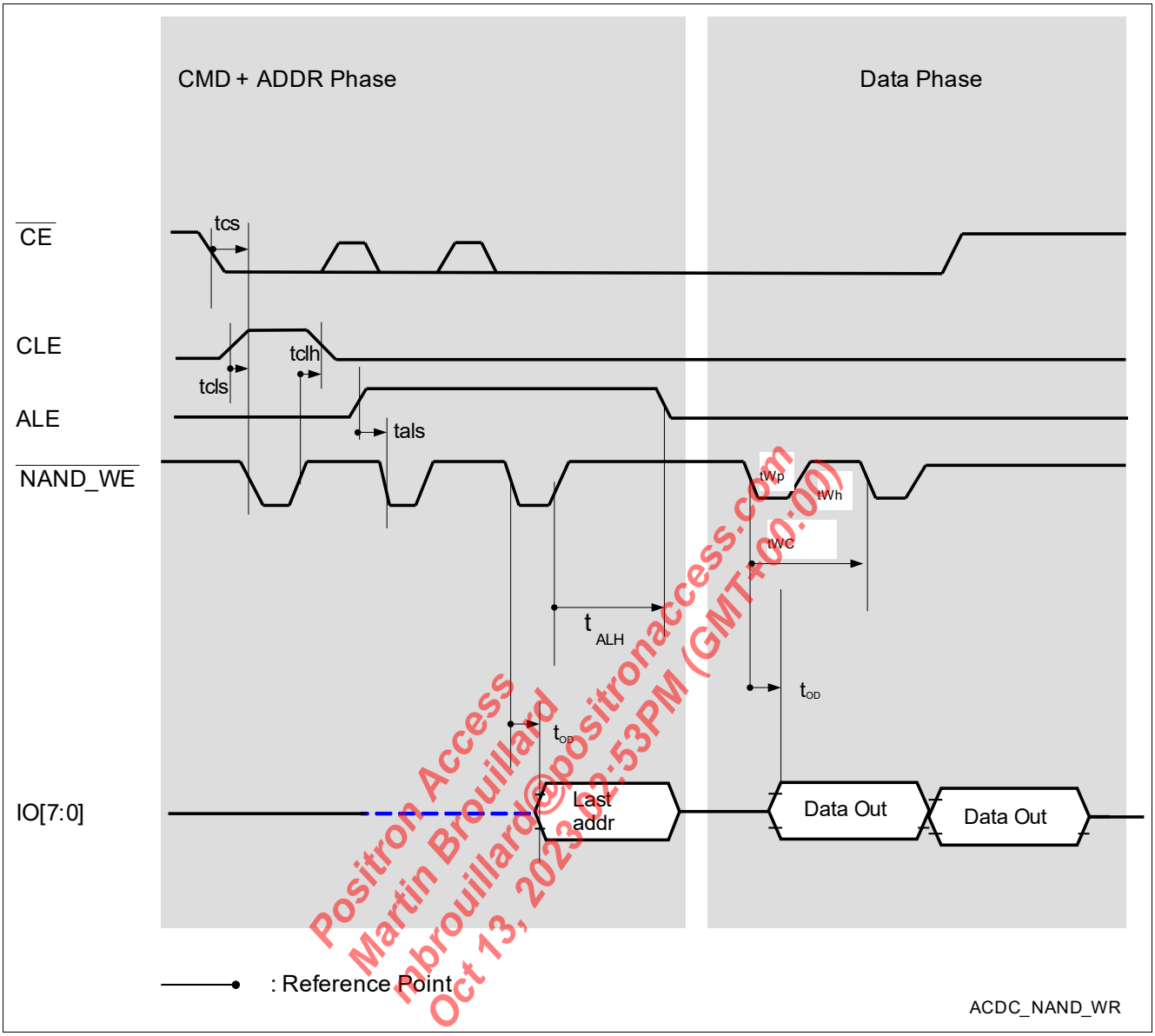


Figure 94 Write Cycle NAND - Pulse Mode

15.5.3 Smart SLIC Interface (SSI) for SSI Timing Interface

This section defines the Smart SLIC Interface timings.

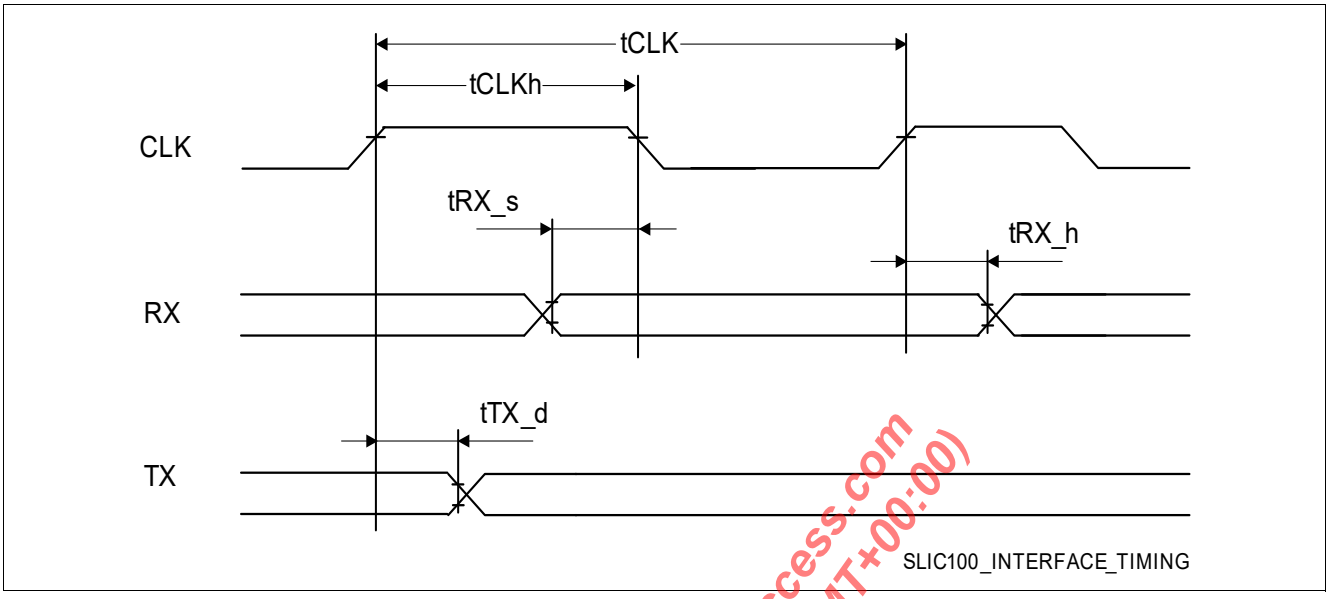


Figure 95 Smart SLIC Interface Timing

Table 154 Timing Values Smart SLIC Interface

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
CLK Period	t_{CLK}	-50 ppm	30.5176	+50 ppm	ns	–
CLK Duty Cycle	t_{CLKh}	–	50	–	%	–
Rx Setup	t_{RX_s}	2	–	–	ns	–
Rx Hold	t_{RX_h}	2	–	–	ns	–
Tx Delay	t_{TX_d}	2[1)]	–	6[2)]	ns	–

1) Best case timing, 5 pF output load.
2) Worst case timing, 30 pF output load. This is equivalent to 12 inch FR4 PCB load plus 33 Ω serial resistor.

15.5.4 TDM Interface Timing

This section shows the timings when the TDM Interface is operating in either Master or Slave mode.

15.5.4.1 PCM - Single-Clocking Mode

Figure 96 shows the single clocking mode.

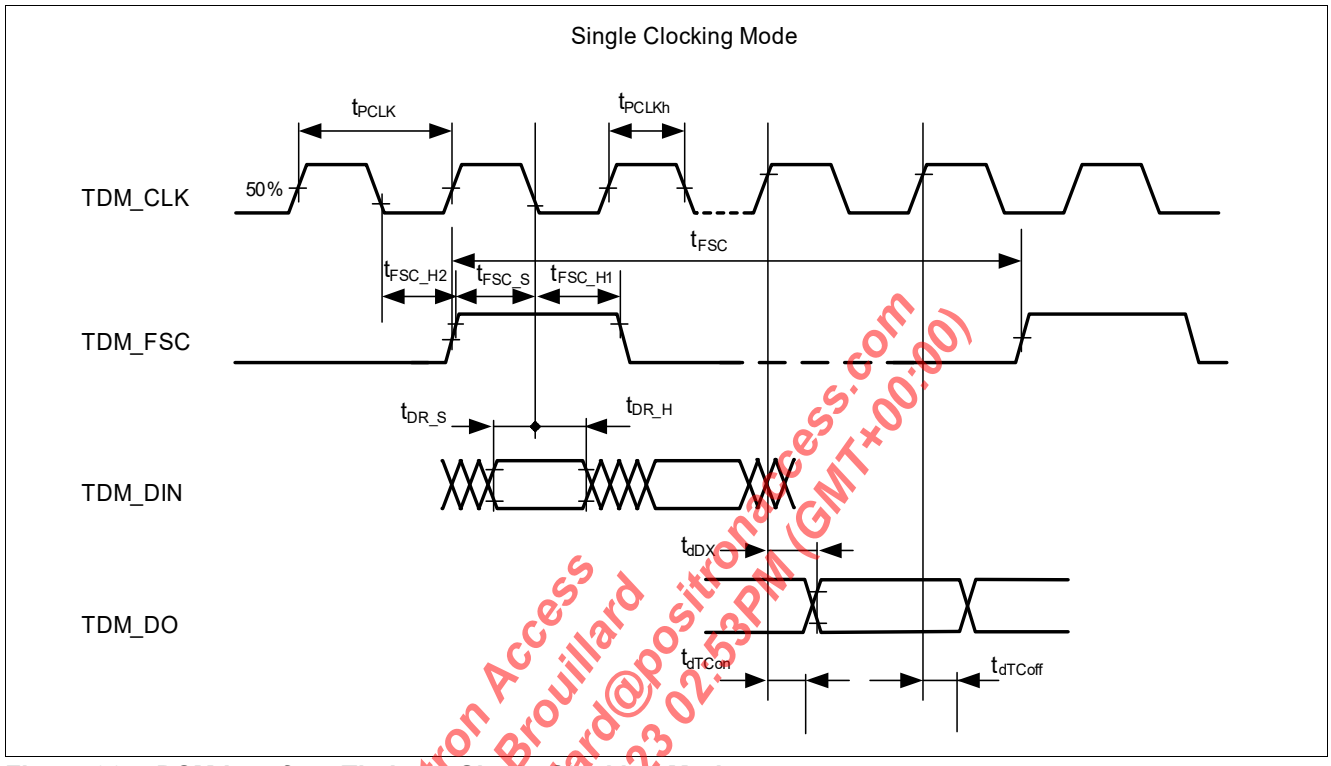


Figure 96 PCM Interface Timing – Single-Clocking Mode

Table 155 Timing Values PCM Interface (Single-Clocking Mode)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Period of P_{CLK} [1]	t_{PCLK}	1/8192	–	1/512	ms	–
P_{CLK} High Time	t_{PCLKh}	0.4 * t_{PCLK}	–	0.6 * t_{PCLK}	µs	–
Period of FSC [1]	t_{FSC}	–	125	–	µs	–
FSC Setup Time	t_{FSC_S}	10	–	–	ns	–
FSC Hold Time 1	t_{FSC_H1}	40	–	$t_{FSC} - t_{PCLK} - t_{FSC\_S}$ [2]	ns	–
FSC Hold Time 2	t_{FSC_H2}	40	–	–	ns	–
DR1/2 Setup Time	t_{DR_S}	10	–	–	ns	–
DR1/2 Hold Time	t_{DR_H}	10	–	–	ns	–
DX1/2 Delay Time	t_{dDX}	–	–	20	ns	–
DX1/2_EN Delay Time On	t_{dTCon}	–	–	20	ns	–
DX1/2_EN Delay Time Off	t_{dTCoff}	–	–	20	ns	–

1) The PCLK frequency must be an integer multiple of the FSC frequency (n*64*f_{FSC}, n = 1...16)

2) This is to ensure that the FSC can be sampled low at least for one t_{PCLK} within t_{FSC}.

15.5.4.2 PCM - Double-Clocking Mode

This section defines the double clocking mode timings.

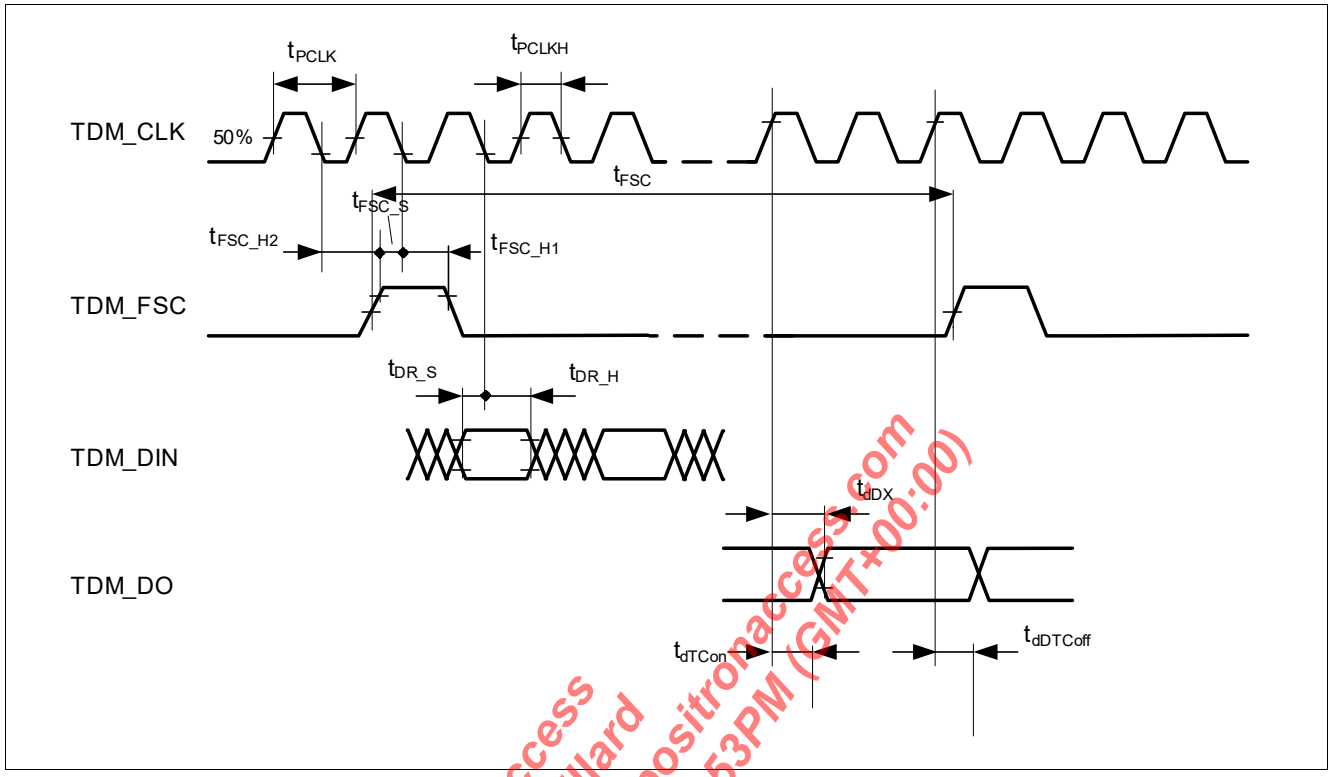


Figure 97 PCM Interface Timing – Double-Clocking Mode

Table 156 Timing Values PCM Interface (Double-Clocking Mode)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Period of P_{CLK} [1]	t_{PCLK}	1/8192	–	1/512	ms	–
P_{CLK} High Time	t_{PCLKh}	0.4 * t_{PCLK}	–	0.6 * t_{PCLK}	µs	–
Period of FSC	t_{FSC}	–	125	–	µs	–
FSC Setup Time	t_{FSC_S}	10	–	–	ns	–
FSC Hold Time 1	t_{FSC_H1}	40	–	$t_{FSC} - t_{PCLK} - t_{FSC\_S}$ [2]	ns	–
FSC Hold Time 2	t_{FSC_H2}	40	–	–	ns	–
DR1/2 Setup Time	t_{DR_S}	10	–	–	ns	–
DR1/2 Hold Time	t_{DR_H}	10	–	–	ns	–
DX1/2 Delay Time	t_{dDX}	–	–	20	ns	–
DX1/2_EN Delay Time On	t_{dTCon}	–	–	20	ns	–
DX1/2_EN Delay Time Off	t_{dTCoff}	–	–	20	ns	–

1) The PCLK frequency must be an integer multiple of the FSC frequency (n*64*f_{FSC}, n = 1...16)

2) This is to ensure that the FSC can be sampled low at least for one t_{PCLK} within t_{FSC}.

15.5.5 PCIe Interface RefCLK and Total Data Jitter

Figure 98 describes the PCIe reference clock.

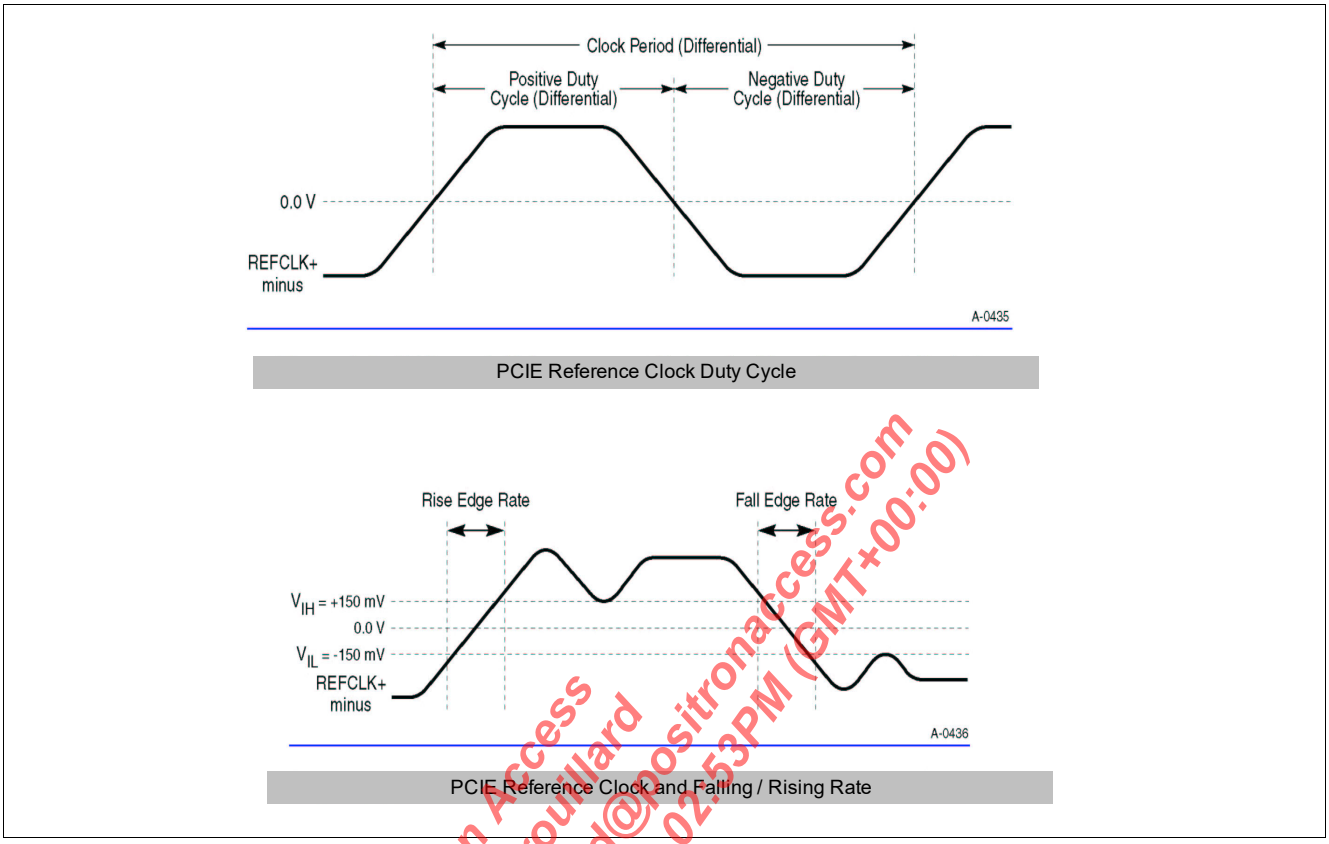


Figure 98 PCIe Clock Timing Diagram

Table 157 PCIe Reference Clock Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
PCI-Ref Clock Period	–	9.847	10	10.203	ns	All specs are met for 100 MHz.
Clock Rising Edge Rate	t_{rise}	0.6	–	0.4	V/ns	-
Clock Falling Edge Rate	t_{fall}	0.6	–	0.4	V/ns	–
Cycle to Cycle Jitter	$t_{ccjitter}$	-	–	150	ps	–
Clock Duty Cycle	-	40	50	60	%	–

Table 158 PCIe Data Jitter for Common Clock Architecture

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Gen 1.0	–	–	–	86	ps, pp	Jitter requirement is after filter function
Gen 2.0	$t_{ccjitter}$	–	–	3.1	ps, RMS	
Gen 3.0	$t_{ccjitter}$	–	–	1.0	ps, RMS	

15.5.6 PCIe 1.x

Table 159 describes the PCIe reference clock and interface eye compliant with v1.1.

Table 159 PCIe 1.1 Transmitter Output Electrical and Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Unit Interval	UI	399.88	400	400.12	ps	Each UI is 400 ps +/-300 ppm. UI does not account for SSC dictated variations.[1)]
Differential Peak to Peak Output Voltage	$V_{TX\text{-}DIFFp\text{-}p}$	0.800	-	1.2	V	$V_{TX\text{-}DIFFp\text{-}p} = 2*\lvert V_{TX\text{-}D+} - V_{TX\text{-}D-}\rvert$[2)]
De-Emphasized Differential Output Voltage Ratio)	$V_{TX\text{-}DE\_RATIO}$	-3.0	-3.5	-4.0	dB	This is the ratio of the $V_{TX\text{-}DIFFp\text{-}p}$ of the second and following bits after a transition divided by the $VT_{X\text{-}DIFFp\text{-}p}$ of the first bit after a transition. [2)]
Minimum Tx Eye Width	$T_{TX\text{-}EYE}$	0.75	-	-	UI	The maximum transmitter jitter can be derived as $T_{TX\text{-}MAX\text{-}JITTER} = 1 - T_{TX\text{-}EYE}$= 0.25 UI. This parameter is measured with the equivalent of a zero jitter reference clock. [2)3)]
Maximum Time between Jitter Median and Maximum Deviation from Median	$T_{TX\text{-}MEDIAN\text{-}to\text{-}MAX\text{-}JITTER}$	-	-	0.125	UI	Jitter is defined as the measurement variation of the crossing points ($V_{TX\text{-}DIFF}$. = 0 V) in relation to recovered TX UI. [2) 3)]
D+/D- Tx Output Rise/Fall Time	$T_{TX\text{-}RISE}$, $T_{TX\text{-}FALL}$	0.125	-	-	UI	See [2)4)]
RMS AC Peak Common Mode Output Voltage	$V_{TX\text{-}CM\text{-}ACp}$	-	-	20	mV	$V_{TX\text{-}CM\text{-}ACp}$ = RMS($\lvert V_{TX\text{-}D+} + VT_{X\text{-}D-}\rvert$/2 – $V_{TX\text{-}CM\text{-}DC}$) See [2)]
Absolute Delta of DC Common Mode Voltage during L0 and Electrical Idle	$V_{TX\text{-}CM\text{-}DC\text{-}ACTIVE\text{-}IDLE\text{-}DELTA}$	0	-	100	mV	$\lvert V_{TX\text{-}CM\text{-}DC\ [during\ L0]} – V_{TX\text{-}CM\text{-}Idle\text{-}DC[During\ Electrical\ Idle.]}\rvert$ <= 100 mV. $V_{TX\text{-}CM\text{-}DC}$ = $DC_{(avg)}$ of $\lvert V_{TX\text{-}D+} + V_{TX\text{-}D-}\rvert$/2 [L0] $V_{TX\text{-}CM\text{-}Idle\text{-}DC}$= $DC_{(avg)}$ of $\lvert V_{TX\text{-}D+} + V_{TX\text{-}D-}\rvert$/2 [Electrical Idle]. See [2)]

Table 159 PCIe 1.1 Transmitter Output Electrical and Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Absolute Delta of DC Common Mode Voltage between D+ and D-	$V_{TX-CM-DC-LINE-DELTA}$	0	-	25	mV	$\|V_{TX-CM-DC-D+} - V_{TX-CM-DC-D-}\| <= 25mV$. $V_{TX-CM-DC-D+} = DC_{(avg)}$ of $\|V_{TX-D+}\|$ + $V_{TX-CM-DC-D-} = DC_{(avg)}$ of $\|V_{TX-D-}\|$ See [2]
Electrical Idle Differential Peak Output Voltage	$V_{TX-IDLE-DIFFp}$	0	-	20	mV	$V_{TX-IDLE-DIFFp} = \|VT_{X-Idle-D+} - V_{Tx-Idle-D-}\| <= 20$ mV See [2]
Voltage Change Allowed during Receiver Detection	$V_{TX-RCV-DETECT}$	-	-	600	mV	The total amount of voltage change that a transmitter may apply to sense whether a low impedance receiver is present.
Tx DC Common Mode Voltage	$V_{TX-DC-CM}$	0	-	3.6	V	The allowed DC common mode voltage under any condition.
Tx Short Circuit Current Limit	$I_{TX-SHORT}$	-	-	90	mA	The total current the transmitter can provide when shorted to its ground.
Minimum Time Spent in Electrical Idle	$T_{TX-IDLE-MIN}$	50	-	-	UI	The minimum time the transmitter must be in electrical idle. Utilized by the receiver when looking for an electrical idle exit after successfully receiving an electrical idle order set.
Maximum Time to Transition to a Valid Electrical Idle after Sending an Electrical Idle Order Set	$T_{TX-IDLE-SET-TO-IDLE}$	-	-	20	UI	The maximum time for the transmitter to transition to electrical idle. Utilized by the receiver when looking for an electrical idle after successfully receiving an electrical idle order set.

Table 159 PCIe 1.1 Transmitter Output Electrical and Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Maximum Time to Transition to Tx Valid Specification after Leaving an Electrical Idle Condition	$T_{TX-IDLE-TO-DIFF-DATA}$	-	-	20	UI	Maximum time to meet all Tx specifications when transitioning from electrical idle to sending differential data. This is considered a debounce time for Tx to meet all Tx specifications after leaving electrical idle.
Differential Return Loss	$RL_{TX-DIFF}$	10	-	-	dB	Measured over 50 MHz to 1.25 GHz
DC Differential Tx Impedance	$Z_{TX-DIFF-DC}$	80	100	120	Ω	Tx DC differential mode low impedance
Lane-to-Lane Output Skew	$L_{TX-SKEW}$	-	-	500 + 2UI	ps	Static skew between any two transmitter lanes with a single link
AC Coupling Capacitor	C_{TX}	75	-	200	nF	All transmitters must be AC coupled. The AC coupling is either required within the media or within the transmitter component itself
Crosslink Random Time-out	$T_{Crosslink}$	0	-	1	ms	This random time helps to resolve conflicts in crosslink configuration be eventually resulting in only one downstream and one upstream port

1) No test load is necessarily associated with this value.
2) Specified at the measurement point into a timing and voltage compliance test load and measured using the clock recovery function
3) A T_{TX-EYE} = 0.70 75 UI provides for a total sum of deterministic and random jitter budget of $T_{TX-JITTER-MAX}$ = 0.25 UI for the transmitter using the clock recovery function. The $T_{TX-EYE-MEDIAN-to-MAX-JITTER}$ specification ensures a jitter distribution in which the median and the maximum deviation from the median is less than half of the total Tx jitter budget using the clock recovery function. The median is not the same as the mean. The jitter median describes the point in time where the number of jitter points on either side is approximately equal as 5 opposed to the averaged time value. This parameter is measured with the equivalent of a zero jitter reference clock. The T_{TX-EYE} measurement is to be met at the target bit error rate. The $T_{TX-EYE-MEDIAN-to-MAX-JITTER}$ is to be met using the compliance pattern at a sample size of 1,000,000 UI.
4) Measured between 20-80% at transmitter package pins into a test load as shown in **Figure 98** for both V_{TX-D+} and V_{TX-D-}.

Table 160 PCIe 1.1 Receiver Electrical and Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Unit Interval	UI	399.88	400	400.12	ps	Each UI is 400 ps +/- 300 ppm. UI does not account for SSC dictated variations.[1)]
Differential Input Peak to Peak Voltage	$V_{RX-DIFFp-p}$	0.175	-	1.2	V	$V_{RX-DIFFp-p} = 2*\|V_{RX-D+} - V_{RX-D-}\|$[2)]
Minimum Receiver Eye Width	T_{RX-EYE}	0.4	-	-	UI	The maximum interconnect media and transmitter jitter that can be tolerated by the receiver can be derived as $T_{RX-MAX-JITTER} = 1 - T_{RX-EYE} = 0.6$ UI.
Maximum Time between Jitter Median and Maximum Deviation from Median	$T_{RX-MEDIAN-to-MAX-JITTER}$	-	-	0.3	UI	Jitter is defined as the measurement variation of the crossing points ($V_{RX-DIFF}$ = 0 V) in relation to recovered TX UI. [2) 3)]
AC Peak Common Mode Input Voltage	$V_{RX-CM-ACp}$	-	-	150	mV	$V_{RX-CM-ACp} = \|V_{RX-D+} + V_{RX-D-}\|/2 - V_{RX-CM-DC})$ VRX-CM-DC = $DC_{(avg)}$ of $\|V_{RX-D+} + V_{RX-D-}\|/2$ See [2)]
Differential Return Loss	$RL_{RX-DIFF}$	10	-	-	dB	Measured over 50 MHz to 1.25 GHz
Common Mode Return Loss	RL_{RX-CM}	6	-	-	dB	Measured over 50 MHz to 1.25 GHz
DC Differential Rx Impedance	$Z_{RX-DIFF-DC}$	80	100	120	Ω	Rx DC differential mode low impedance
DC Input Impedance	Z_{RX-DC}	40	50	60	Ω	Required Rx D+ as well as D- DC impedance (50 Ω +/- 20% tolerance).
Powered Down DC Input Impedance	$Z_{RX-HIGH-IMP-DC}$	50 k	-	-	Ω	Required Rx D+ as well as D- DC impedance when the receiver terminations do not have power. See[3)]
Electrical Idle Detect Threshold	$V_{RX-IDLE-DET-DIFFp-p}$	65	-	175	mV	$V_{RX-IDLE-DET-DIFFp-p} = 2*\|V_{RX-D+} - V_{RX-D-}\|$

Table 160 PCIe 1.1 Receiver Electrical and Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Unexpected Electrical Idle Enter Detect Threshold Integration Time	$V_{RX-IDLE-DET-DIFF-ENTERTIME}$		-	10	ms	An unexpected electrical idle ($V_{RX-DIFFp-p} < V_{RX-IDLE-DET-DIFFp-p)}$) must be recognized no longer than $T_{RX-IDLE-DET-DIFF-ENTERTIME}$ to signal an unexpected idle condition.
Total Skew	$L_{RX-SKEW}$	-	-	20	ns	Skew across all lanes on a link. This includes variation in the length of a SKP ordered set (for example, COM and one to five SKP symbols) at the Rx as well as any delay differences arising from the interconnect itself.

1) No test load is necessarily associated with this value.
2) Specified at the measurement point and measured using the clock recovery function. The test load in **Figure 99** must be used as the Rx device when taking measurements (also refer to the receiver compliance eye diagram shown in **Figure 100**). When the clocks to the Rx and Tx are not derived from the same reference clock, the TX UI recovered using the clock recovery function must be used as a reference for the eye diagram.
3) Impedance during all LTSSM states. When transitioning from a Fundamental Reset to Detect (the 30 initial state of the LTSSM) there is a 5 ms transition time before receiver termination values must be met on all unconfigured lanes of a port.

Figure 101 shows the PCIe eye diagram must comply with PCIe Base Specification v1.1.

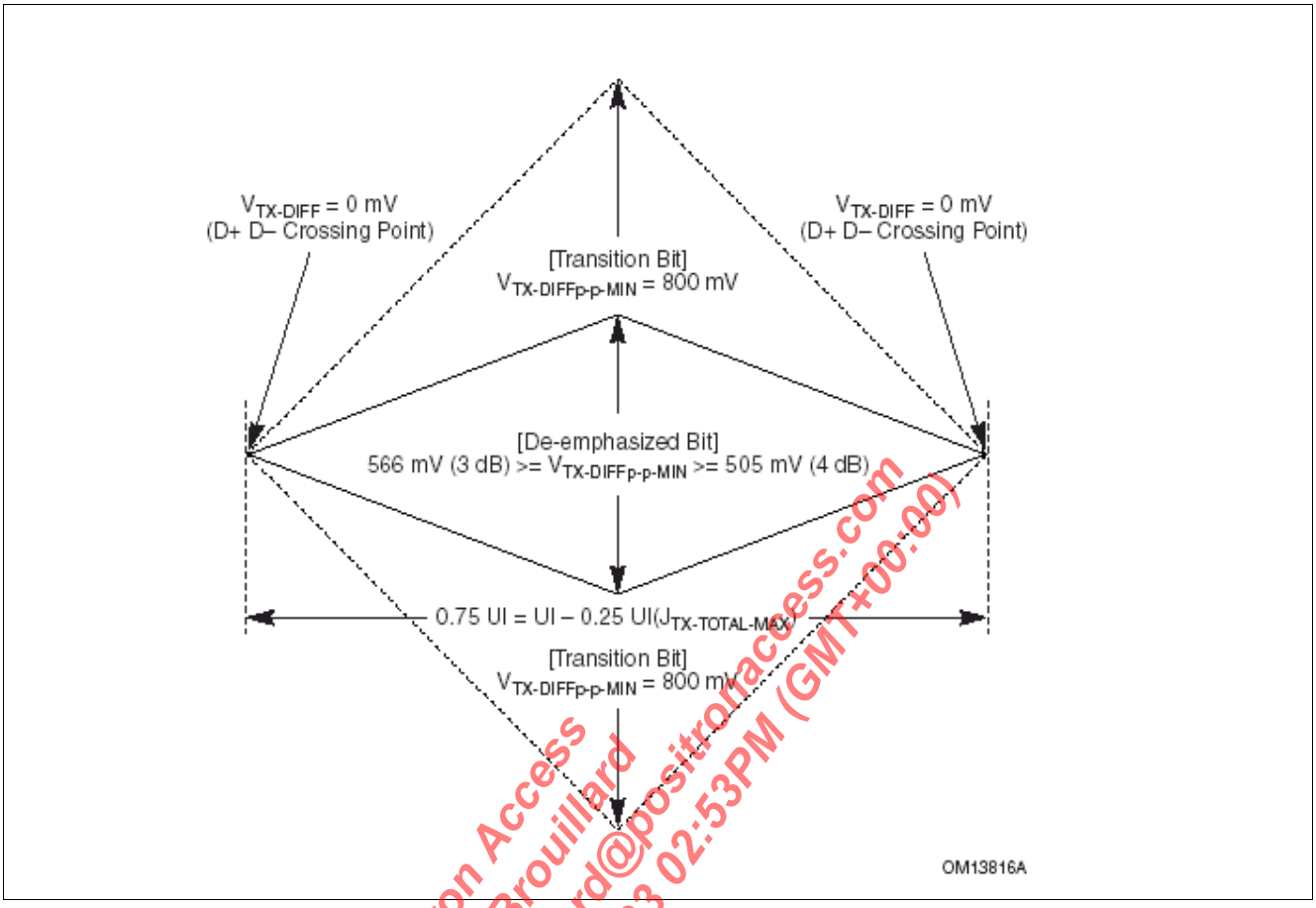


Figure 99 PCIe v1.1 Eye Diagram Compliance (at Tx Ball)

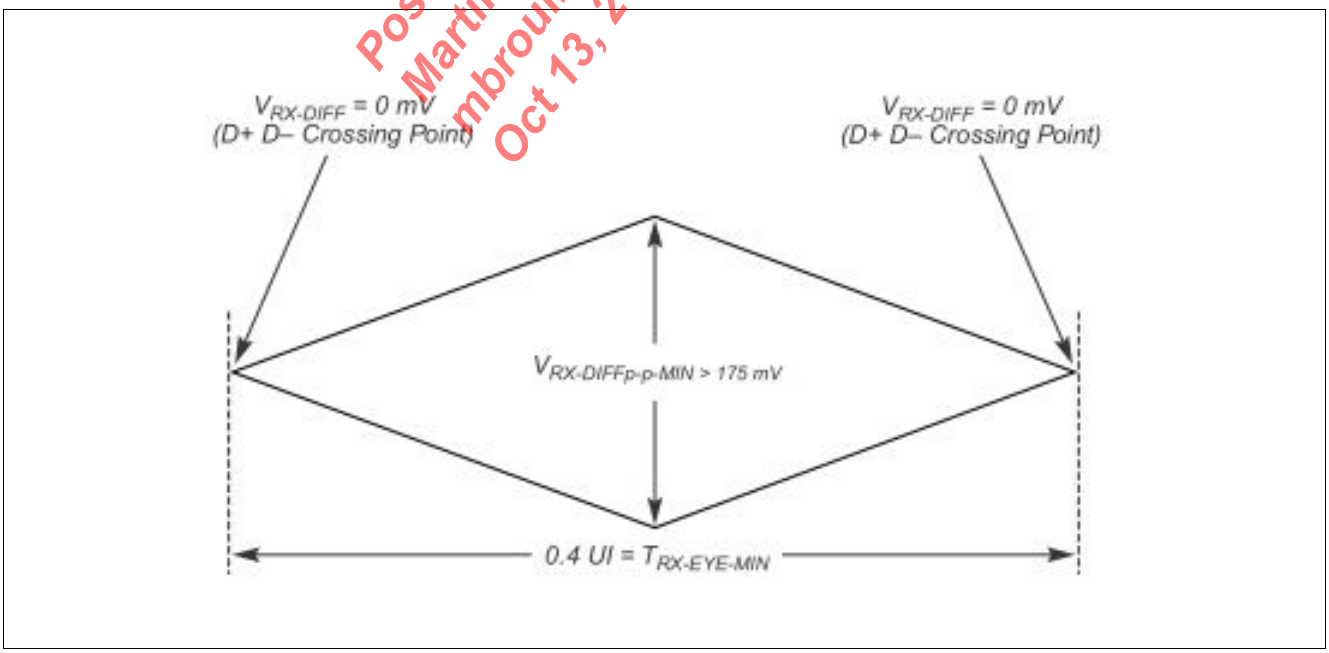


Figure 100 PCIe v1.1 Eye Diagram Compliance (at Rx Ball)

The PCIe test/measurement load must comply with the PCIe Base Specification v1.1 as shown in **Figure 101**.

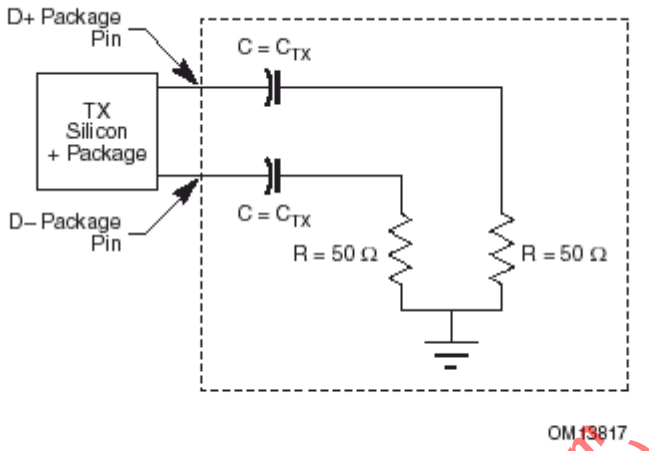


Figure 101 PCIe v1.1 Test/Measurement Load Compliance

15.5.7 PCIe 2.0

Table 161 PCIe 2.0 Transmitter Output Electrical and Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Unit Interval	*UI*	199.94	200	200.06	ps	Each UI is 200 ps +/-300 ppm. UI does not account for SSC dictated variations.[1]
Differential Peak to Peak Output Voltage	$V_{TX\text{-}DIFFp\text{-}p}$	0.800	-	1.2	V	$V_{TX\text{-}DIFFp\text{-}p} = 2*\lvert V_{TX\text{-}D}+ - V_{TX\text{-}D}-\rvert$[2]
Differential Peak to Peak Output Voltage, Low Power Mode	$V_{TX\text{-}DIFFp\text{-}p}$	0.400	-	1.2	V	$V_{TX\text{-}DIFFp\text{-}p} = 2*\lvert V_{TX\text{-}D}+ - V_{TX\text{-}D}-\rvert$[2]
De-Emphasized Differential Output Voltage Ratio)	$V_{TX\text{-}DE\_RATIO}$	-3.0	-3.5	-4.0	dB	This is the ratio of the $V_{TX\text{-}DIFFp\text{-}p}$ of the second and following bits after a transition divided by the $VT_{X\text{-}DIFFp\text{-}p}$ of the first bit after a transition. [2]
Minimum TX Eye Width	$T_{TX\text{-}EYE}$	0.75	-	-	UI	The maximum transmitter jitter can be derived as $T_{TX\text{-}MAX\text{-}JITTER} = 1 - T_{TX\text{-}EYE} = 0.25$ UI. This parameter is measured with the equivalent of a zero jitter reference clock. [2][3]
D+/D- TX Output Rise/Fall Time	$T_{TX\text{-}RISE}$, $T_{TX\text{-}FALL}$	0.15	-	-	UI	See [2][4]

Table 161 PCIe 2.0 Transmitter Output Electrical and Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
AC Peak Common Mode Output Voltage	$V_{TX-CM-ACp}$	-	-	100	mV	-
Absolute Delta of DC Common Mode Voltage during L0 and Electrical Idle	$V_{TX-CM-DC-ACTIVE-IDLE-DELTA}$	0	-	100	mV	$\|V_{TX-CM-DC\ [during\ L0]} - V_{TX-CM-Idle-DC[During\ Electrical\ Idle.]}\|$ <= 100 mV. $V_{TX-CM-DC}$ = $DC_{(avg)}$ of $\|V_{TX-D+} + V_{TX-D-}\|/2$ [L0] $V_{TX-CM-Idle-DC}$= $DC_{(avg)}$ of $\|V_{TX-D+} + V_{TX-D-}\|/2$ [Electrical Idle] See 2)
Absolute Delta of DC Common Mode Voltage between D+ and D-	$V_{TX-CM-DC-LINE-DELTA}$	0	-	25	mV	$\|V_{TX-CM-DC-D+} - V_{TX-CM-DC-D-}$ <= 25 mV. $V_{TX-CM-DC-D+}$ = $DC_{(avg)}$ of $\|V_{TX-D+} + V_{TX-CM-DC-D-}$ = $DC_{(avg)}$ of $\|V_{TX-D-}\|$ See 2)
Electrical Idle Differential Peak Output Voltage	$V_{TX-IDLE-DIFFp}$	0	-	20	mV	$V_{TX-IDLE-DIFFp}$ = $\|VT_{X-Idle-D+} - V_{Tx-Idle-D-}\|$ <= 20 mV See 2)
Electrical Idle Differential Output Voltage	$V_{TX-IDLE-DIFFDC}$	0	-	5	mV	$V_{TX-IDLE-DIFFp}$ = $\|VT_{X-Idle-D+} - V_{Tx-Idle-D-}\|$ <= 5 mV
Voltage Change Allowed During Receiver Detection	$V_{TX-RCV-DETECT}$	-	-	600	mV	The total amount of voltage change that a transmitter may apply to sense whether a low impedance receiver is present.
TX DC Common Mode Voltage	$V_{TX-DC-CM}$	0	-	3.6	V	The allowed DC common mode voltage under any condition.
TX Short Circuit Current Limit	$I_{TX-SHORT}$	-	-	90	mA	The total current the transmitter can provide when shorted to its ground.
Minimum Time Spent in Electrical Idle	$T_{TX-IDLE-MIN}$	20	-	-	ns	The minimum time the transmitter must be in electrical idle. Utilized by the receiver when looking for an electrical idle exit after successfully receiving an electrical idle order set.
Maximum Time to Transition to a Valid Electrical Idle after Sending an Electrical Idle Order Set	$T_{TX-IDLE-SET-TO-IDLE}$	-	-	8	ns	The maximum time for the transmitter to transition to electrical idle. Utilized by the receiver when looking for an electrical idle after successfully receiving an electrical idle order set.

Table 161 PCIe 2.0 Transmitter Output Electrical and Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Maximum Time to Transition to TX Valid Specification after Leaving an Electrical Idle Condition	$T_{TX-IDLE-TO-DIFF-DATA}$	-	-	8	ns	Maximum time to meet all TX specifications when transitioning from electrical idle to sending differential data. This is considered a debounce time for TX to meet all TX specifications after leaving electrical idle.
Differential Return Loss	$RL_{TX-DIFF}$	10	-	-	dB	Measured over 50 MHz to 1.25 GHz
Differential Return Loss	$RL_{TX-DIFF}$	8	-	-	dB	Measured over 1.25G MHz to 2.5 GHz
DC Differential TX Impedance	$Z_{TX-DIFF-DC}$	-	-	120	Ω	TX DC differential mode low impedance
Lane-to-Lane Output Skew	$L_{TX-SKEW}$	-	-	500 + 4UI	ps	Static skew between any two transmitter lanes with a single link
AC Coupling Capacitor	C_{TX}	75	-	200	nF	All transmitters must be AC coupled. The AC coupling is either required within the media or within the transmitter component itself.
Crosslink Random Time-out	$T_{Crosslink}$	0		1	ms	This random time helps to resolve conflicts in crosslink configuration be eventually resulting in only one downstream and one upstream port.

1) No test load is necessarily associated with this value.
2) Specified at the measurement point into a timing and voltage compliance test load and measured using the clock recovery function
3) A T_{TX-EYE} = 0.70 75 UI provides for a total sum of deterministic and random jitter budget of $T_{TX-JITTER-MAX}$ = 0.25 UI for the transmitter using the clock recovery function. The $T_{TX-EYE-MEDIAN-to-MAX-JITTER}$ specification ensures a jitter distribution in which the median and the maximum deviation from the median is less than half of the total TX jitter budget using the clock recovery function. The median is not the same as the mean. The jitter median describes the point in time where the number of jitter points on either side is approximately equal as 5 opposed to the averaged time value. This parameter is measured with the equivalent of a zero jitter reference clock. The T_{TX-EYE} measurement is to be met at the target bit error rate. The $T_{TX-EYE-MEDIAN-to-MAX-JITTER}$ is to be met using the compliance pattern at a sample size of 1,000,000 UI.
4) Measured between 20-80% at transmitter package pins into a test load as shown in **Figure 102** for both V_{TX-D+} and V_{TX-D-}.

Table 162 PCIe 2.0 Receiver Electrical and Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Unit Interval	UI	199.94	200	200.06	ps	Each UI is 200 ps +/- 300 ppm. UI does not account for SSC dictated variations.[1)]
Differential Input Peak to Peak Voltage	$V_{RX\text{-}DIFFp\text{-}p}$	0.12	-	1.2	V	$V_{RX\text{-}DIFFp\text{-}p} = 2*\lvert V_{RX\text{-}D+} - V_{RX\text{-}D-}\rvert$[2)]
Minimum Receiver Eye Width	$T_{RX\text{-}EYE}$	-	-	-	UI	Not specified
Maximum Time Between Jitter Median and Maximum Deviation from Median	$T_{RX\text{-}MEDIAN\text{-}to\text{-}MAX\text{-}JITTER}$	-	-	-	UI	Not specified
AC Peak Common Mode Input Voltage	$V_{RX\text{-}CM\text{-}ACp}$	-	-	150	mV	$V_{RX\text{-}CM\text{-}ACp} = \lvert V_{RX\text{-}D+} + V_{RX\text{-}D-}\rvert/2 - V_{RX\text{-}CM\text{-}DC}$) VRX-CM-DC = $DC_{(avg)}$ of $\lvert V_{RX\text{-}D+} + V_{RX\text{-}D-}\rvert/2$ See [2)]
Differential Return Loss	$RL_{RX\text{-}DIFF}$	10	-	-	dB	Measured over 50 MHz to 1.25 GHz
Differential Return Loss	$RL_{RX\text{-}DIFF}$	8	-	-	dB	Measured over 1.25 GHz to 2.5 GHz
Common Mode Return Loss	$RL_{RX\text{-}CM}$	6	-	-	dB	Measured over 50 MHz to 1.25 GHz
DC Input Impedance	$Z_{RX\text{-}DC}$	40	50	60	Ω	Required Rx D+ as well as D- DC impedance (50 Ω +/- 20% tolerance).
Powered Down DC Input Impedance	$Z_{RX\text{-}HIGH\text{-}IMP\text{-}DC}$	50 k	-	-	Ω	Required Rx D+ as well as D- DC impedance when the receiver terminations do not have power. See[3)]
Electrical Idle Detect Threshold	$V_{RX\text{-}IDLE\text{-}DET\text{-}DIFFp\text{-}p}$	65	-	175	mV	$V_{RX\text{-}IDLE\text{-}DET\text{-}DIFFp\text{-}p} = 2*\lvert V_{RX\text{-}D+} - V_{RX\text{-}D-}\rvert$

Table 162 PCIe 2.0 Receiver Electrical and Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Unexpected Electrical Idle Enter Detect Threshold Integration Time	$V_{RX-IDLE-DET-DIFF-ENTERTIME}$		-	10	ms	An unexpected electrical idle ($V_{RX-DIFFp-p} < V_{RX-IDLE-DET-DIFFp-p)}$) must be recognized no longer than $T_{RX-IDLE-DET-DIFF-ENTERTIME}$ to signal an unexpected idle condition.
Total Skew	$L_{RX-SKEW}$	-	-	8	ns	Skew across all lanes on a link. This includes variation in the length of a SKP ordered set (for example, COM and one to five SKP symbols) at the Rx as well as any delay differences arising from the interconnect itself.

1) No test load is necessarily associated with this value.
2) Specified at the measurement point and measured using the clock recovery function. The test load in **Figure 102** must be used as the Rx device when taking measurements (also refer to the receiver compliance eye diagram shown in **Figure 104**). When the clocks to the Rx and Tx are not derived from the same reference clock, the TX UI recovered using the clock recovery function must be used as a reference for the eye diagram.
3) Impedance during all LTSSM states. When transitioning from a Fundamental Reset to Detect (the 30 initial state of the LTSSM) there is a 5 ms transition time before receiver termination values must be met on all unconfigured lanes of a port.

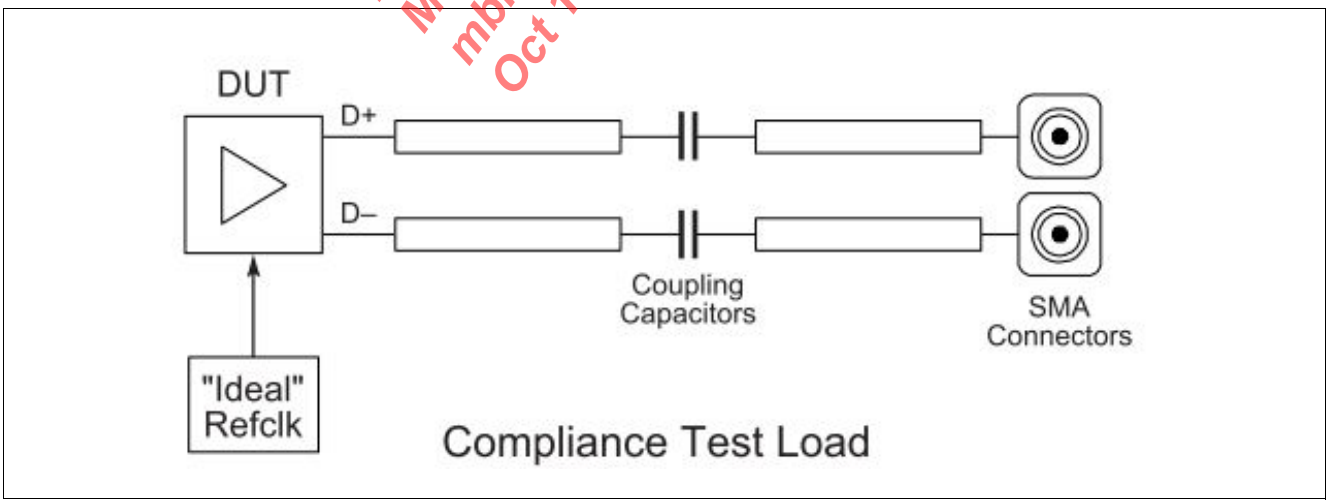


Figure 102 PCIe 2.0 Test/Measurement Load Compliance

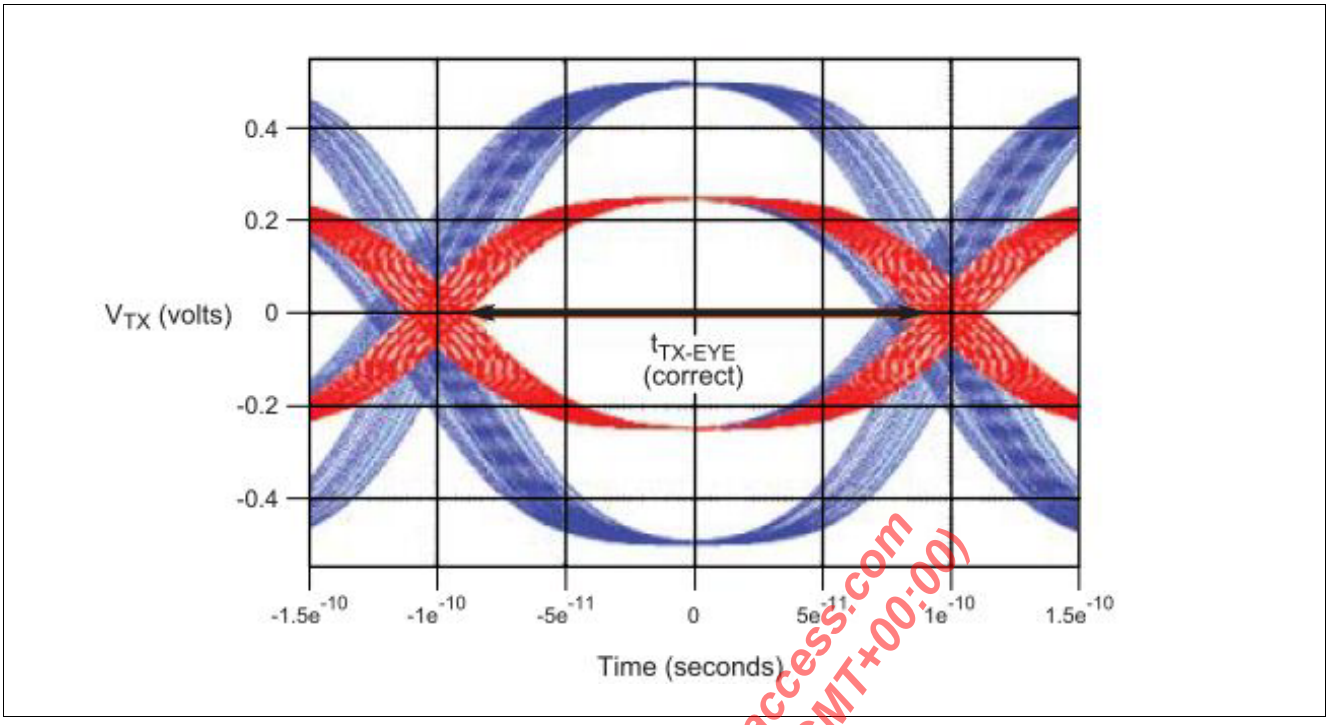


Figure 103 PCIe 2.0 Eye Diagram Compliance (at Tx ball, with De-emphasis Jitter Removal)

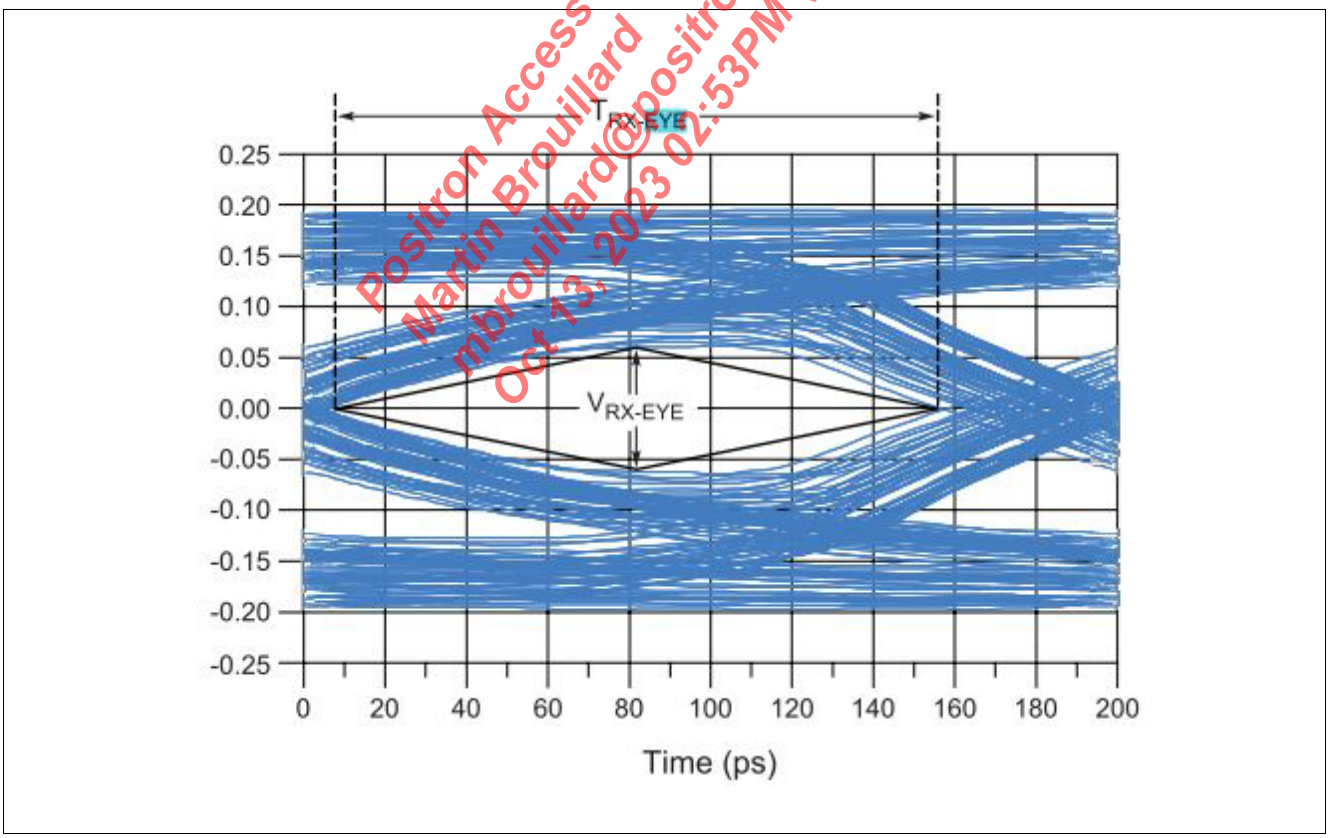


Figure 104 PCIe 2.0 Eye Diagram Compliance (at Rx ball)

15.5.8 PCIe 3.0

Table 163 PCIe 3.0 Transmitter Output Electrical and Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Unit Interval	UI	124.9625	125	125.0375	ps	Each UI is 125 ps +/- 300 ppm. UI does not account for SSC dictated variations.[1]
Differential Peak to Peak Output Voltage	$V_{TX-DIFFp-p}$	0.800	-	1.3	V	$V_{TX-DIFFp-p}$ = 2*|V_{TX-D}+ - V_{TX-D}-|[2]
De-emphasized Differential Output Voltage Ratio)	V_{TX-DE_RATIO}	-3.0	-3.5	-4.0	dB	This is the ratio of the $V_{TX-DIFFp-p}$ of the second and following bits after a transition divided by the $VT_{X-DIFFp-p}$ of the first bit after a transition. [2]
Minimum Tx Eye Width	T_{TX-EYE}	0.75	-	-	UI	The maximum transmitter jitter can be derived as $T_{TX-MAX-JITTER}$ = 1 - T_{TX-EYE}= 0.25 UI. This parameter is measured with the equivalent of a zero jitter reference clock. [2][3]
D+/D- Tx Output Rise/Fall Time	$T_{TX-RISE}$, $T_{TX-FALL}$	0.15	-	-	UI	See [2][4]
AC Peak Common Mode Output Voltage	$V_{TX-CM-ACp}$	-	-	100	mV	-
Absolute Delta of DC Common Mode Voltage during L0 and Electrical Idle	$V_{TX-CM-DC-ACTIVE-IDLE-DELTA}$	0	-	100	mV	|$V_{TX-CM-DC\ [during\ L0]}$ – $V_{TX-CM-Idle-DC[During\ Electrical\ Idle.]}$| <= 100mV. $V_{TX-CM-DC}$ = $DC_{(avg)}$ of |V_{TX-D+} + V_{TX-D-}|/2 [L0] $V_{TX-CM-Idle-DC}$= $DC_{(avg)}$ of |V_{TX-D+} + V_{TX-D-}|/2 [Electrical Idle] See [2]
Absolute Delta of DC Common Mode Voltage between D+ and D-	$V_{TX-CM-DC-LINE-DELTA}$	0	-	25	mV	|$V_{TX-CM-DC-D+}$ – $V_{TX-CM-DC-D-}$<= 25mV. $V_{TX-CM-DC-D+}$ = $DC_{(avg)}$ of |V_{TX-D+} + $V_{TX-CM-DC-D-}$ = $DC_{(avg)}$ of |V_{TX-D-}| See [2]
Electrical Idle Differential Peak Output Voltage	$V_{TX-IDLE-DIFFp}$	0	-	20	mV	$V_{TX-IDLE-DIFFp}$ = |$VT_{X-Idle-D+}$ - $V_{Tx-Idle-D-}$| <= 20 mV See [2]

Table 163 PCIe 3.0 Transmitter Output Electrical and Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Electrical Idle Differential Output Voltage	$V_{TX\text{-}IDLE\text{-}DIFFDC}$	0	-	5	mV	$V_{TX\text{-}IDLE\text{-}DIFFp}$ = |$VT_{X\text{-}Idle\text{-}D+}$ - $V_{Tx\text{-}Idle\text{-}D-}$| <= 5 mV
Amount of Voltage Change Allowed during Receiver Detection	$V_{TX\text{-}RCV\text{-}DETECT}$	-	-	600	mV	The total amount of voltage change that a transmitter may apply to sense whether a low impedance receiver is present.
Tx DC Common Mode Voltage	$V_{TX\text{-}DC\text{-}CM}$	0	-	3.6	V	The allowed DC common mode voltage under any condition.
Tx Short Circuit Current Limit	$I_{TX\text{-}SHORT}$	-	-	90	mA	The total current the transmitter can provide when shorted to its ground.
Minimum Time Spent in Electrical Idle	$T_{TX\text{-}IDLE\text{-}MIN}$	20	-	-	ns	The minimum time the transmitter must be in electrical idle. Utilized by the receiver when looking for an electrical idle exit after successfully receiving an electrical idle order set.
Maximum Time to Transition to a Valid Electrical Idle after Sending an Electrical Idle Order Set	$T_{TX\text{-}IDLE\text{-}SET\text{-}TO\text{-}IDLE}$	-	-	8	ns	The maximum time for the transmitter to transition to electrical idle. Utilized by the receiver when looking for an electrical idle after successfully receiving an electrical idle order set.
Maximum Time to Transition to Tx Valid Specification after Leaving an Electrical Idle Condition	$T_{TX\text{-}IDLE\text{-}TO\text{-}DIFF\text{-}DATA}$	-	-	8	ns	Maximum time to meet all Tx specifications when transitioning from electrical idle to sending differential data. This is considered a debounce time for Tx to meet all Tx specifications after leaving electrical idle.
Differential Return Loss	$RL_{TX\text{-}DIFF}$	10	-	-	dB	Measured over 50 MHz to 1.25 GHz

Table 163 PCIe 3.0 Transmitter Output Electrical and Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Differential Return Loss	$RL_{TX-DIFF}$	8	-	-	dB	Measured over 1.25G MHz to 2.5 GHz
DC Differential Tx Impedance	$Z_{TX-DIFF-DC}$	-	-	120	Ω	Tx DC differential mode low impedance
Lane-to-Lane Output Skew	$L_{TX-SKEW}$	-	-	500 + 4UI	ps	Static skew between any two transmitter lanes with a single link
AC Coupling Capacitor	C_{TX}	75	-	220	nF	All transmitters must be AC coupled. The AC coupling is either required within the media or within the transmitter component itself.
Crosslink Random Time-out	$T_{Crosslink}$	0	-	1	ms	This random time helps to resolve conflicts in crosslink configuration be eventually resulting in only one downstream and one upstream port.

1) No test load is necessarily associated with this value.
2) Specified at the measurement point into a timing and voltage compliance test load and measured using the clock recovery function
3) A T_{TX-EYE} = 0.70 75 UI provides for a total sum of deterministic and random jitter budget of $T_{TX-JITTER-MAX}$ = 0.25 UI for the transmitter using the clock recovery function. The $T_{TX-EYE-MEDIAN-to-MAX-JITTER}$ specification ensures a jitter distribution in which the median and the maximum deviation from the median is less than half of the total TX jitter budget using the clock recovery function. The median is not the same as the mean. The jitter median describes the point in time where the number of jitter points on either side is approximately equal as 5 opposed to the averaged time value. This parameter is measured with the equivalent of a zero jitter reference clock. The T_{TX-EYE} measurement is to be met at the target bit error rate. The $T_{TX-EYE-MEDIAN-to-MAX-JITTER}$ is to be met using the compliance pattern at a sample size of 1,000,000 UI.
4) Measured between 20-80% at transmitter package pins into a test load as shown in **Figure 102** for both V_{TX-D+} and V_{TX-D-}.

Table 164 PCIe 3.0 Receiver Electrical and Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Unit Interval	UI	124.9625	125	125.0375	ps	Each UI is 125 ps +/-300 ppm. UI does not account for SSC dictated variations.[1)]
Differential Input Peak to Peak Voltage	$V_{RX-DIFFp-p}$	0.12	-	1.3	V	$V_{RX-DIFFp-p} = 2*\|V_{RX-D+} - V_{RX-D-}\|$[2)]
Minimum Receiver Eye Width	T_{RX-EYE}	0.30	-	0.35	UI	-.
Maximum Time between Jitter Median and Maximum Deviation from Median	$T_{RX-MEDIAN-to-MAX-JITTER}$	0.30	-	-	UI	-
AC Peak Common Mode Input Voltage	$V_{RX-CM-ACp}$	-	-	75	mV	$V_{RX-CM-ACp} = \|V_{RX-D+} + V_{RX-D-}\|/2 – V_{RX-CM-DC}$) VRX-CM-DC = $DC_{(avg)}$ of $\|V_{RX-D+} + V_{RX-D-}\|/2$ See [2)]
Differential Return Loss	$RL_{RX-DIFF}$	-10	-	-8	dB	Measured over 50 MHz to 1.25 GHz
Differential Return Loss	$RL_{RX-DIFF}$	-8	-	-6	dB	Measured over 1.25 GHz to 4 GHz
Common Mode Return Loss	RL_{RX-CM}	-6	-	-5	dB	Measured over 50 MHz to 4 GHz
DC Input Impedance	Z_{RX-DC}	40	50	60	Ω	Required Rx D+ as well as D- DC impedance (50 Ω +/- 20% tolerance).
Powered Down DC Input Impedance	$Z_{RX-HIGH-IMP-DC}$	20 k	-	-	Ω	Required Rx D+ as well as D- DC impedance when the receiver terminations do not have power. See[3)]
Electrical Idle Detect Threshold	$V_{RX-IDLE-DET-DIFFp-p}$	65	-	175	mV	$V_{RX-IDLE-DET-DIFFp-p} = 2*\|V_{RX-D+} - V_{RX-D-}\|$

Table 164 PCIe 3.0 Receiver Electrical and Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Unexpected Electrical Idle Enter Detect Threshold Integration Time	$V_{RX-IDLE-DET-DIFF-ENTERTIME}$		-	10	ms	An unexpected electrical idle ($V_{RX-DIFFp-p} < V_{RX-IDLE-DET-DIFFp-p)}$) must be recognized no longer than $T_{RX-IDLE-DET-DIFF-ENTERTIME}$ to signal an unexpected idle condition.
Total Skew	$L_{RX-SKEW}$	-	-	6	ns	Skew across all lanes on a link. This includes variation in the length of a SKP ordered set (for example, COM and one to five SKP symbols) at the Rx as well as any delay differences arising from the interconnect itself.

1) No test load is necessarily associated with this value.
2) Specified at the measurement point and measured using the clock recovery function. The test load in **Figure 102** must be used as the Rx device when taking measurements (also refer to the receiver compliance eye diagram shown in **Figure 104**). When the clocks to the Rx and Tx are not derived from the same reference clock, the TX UI recovered using the clock recovery function must be used as a reference for the eye diagram.
3) Impedance during all LTSSM states. When transitioning from a Fundamental Reset to Detect (the 30 initial state of the LTSSM) there is a 5 ms transition time before receiver termination values must be met on all unconfigured lanes of a port.

15.5.9 PCIe 4.0

Based on version 0.9 of the PCIe revision 4.0 specification.

Table 165 PCIe 4.0 Transmitter Output Electrical and Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Unit Interval	UI	62.481225	62.5	62.51875	ps	Each UI is 62.5 ps +/-300 ppm. UI does not account for SSC dictated variations.[1)]
Differential Peak-to-peak Output Voltage	$V_{TX-DIFFp-p}$	0.800	-	1.3	V	$V_{TX-DIFFp-p} = 2*\|V_{TX-D}+ - V_{TX-D}-\|$[2)]
AC Peak Common Mode Output Voltage	V_{TX-AC_CM-PP}	-	-	150	mV	-
Absolute Delta of DC Common Mode Voltage during L0 and Electrical Idle	$V_{TX-CM-DC-ACTIVE-IDLE-DELTA}$	0	-	100	mV	$\|V_{TX-CM-DC\ [during\ L0]} – V_{TX-CM-Idle-DC[During\ Electrical\ Idle.]}\| <=$ 100mV. $V_{TX-CM-DC\ =}\ DC_{(avg)}$ of $\|V_{TX-D+} + V_{TX-D-}\|/2$ [L0] $V_{TX-CM-Idle-DC}= DC_{(avg)}$ of $\|V_{TX-D+} + V_{TX-D-}\|/2$ [Electrical Idle] See [2)]
Absolute Delta of DC Common Mode Voltage between D+ and D-	$V_{TX-CM-DC-LINE-DELTA}$	0	-	25	mV	$\|V_{TX-CM-DC-D+} – V_{TX-CM-DC-D-}\| <=$ 25mV. $V_{TX-CM-DC-D+} = DC_{(avg)}$ of $\|V_{TX-D+} + V_{TX-CM-DC-D-} = DC_{(avg)}$ of $\|V_{TX-D-}\|$ See [2)]
Electrical Idle Differential Peak Output Voltage	$V_{TX-IDLE-DIFFACp}$	0	-	20	mV	$V_{TX-IDLE-DIFFp} = \|VT_{X-Idle-D+} - V_{Tx-Idle-D-}\| <=$ 20 mV See [2)]
Electrical Idle Differential Output Voltage	$V_{TX-IDLE-DIFFDC}$	0	-	5	mV	$V_{TX-IDLE-DIFFp} = \|VT_{X-Idle-D+} - V_{Tx-Idle-D-}\| <=$ 5 mV
Amount of Voltage Change Allowed during Receiver Detection	$V_{TX-RCV-DETECT}$	-	-	600	mV	Total amount of voltage change a Transmitter is able to apply to sense whether a low impedance receiver is present.
TX DC Peak-to-peak Common Mode Voltage	$V_{TX-DC-CM}$	0	-	3.6	V	Allowed DC common mode voltage under any condition.
TX Short Circuit Current Limit	$I_{TX-SHORT}$	-	-	90	mA	Total current the transmitter is able to provide when shorted to its ground.

Table 165 PCIe 4.0 Transmitter Output Electrical and Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Minimum time spend in electrical idle	$T_{TX-IDLE-MIN}$	20	-	-	ns	The minimum time the transmitter must be in Electrical Idle utilized by the receiver to start looking for an electrical idle Exit after successful receiving and Electrical idle order set.
Maximum time to transition to a valid electrical idle after sending an Electrical idle order set	$T_{TX-IDLE-SET-TO-IDLE}$	-	-	8	ns	The minimum time the transmitter must be in Electrical Idle utilized by the receiver to start looking for an electrical idle Exit after successful receiving and Electrical idle order set.
Maximum time to transition to TX valid specification after leaving an electrical idle condition	$T_{TX-IDLE-TO-DIFF-DATA}$	-	-	8	ns	Maximum time to meet all TX specifications when transitioning from Electrical idle to sending differential data. This is considered a debounce time for TX to meet all TX specifications after leaving Electrical Idle.
Differential Return Loss	$RL_{TX-DIFF}$	-10	-	-6	dB	Measured over 0.05G MHz to 8 GHz
DC Differential TX Impedance	$Z_{TX-DIFF-DC}$	-	-	120	Ω	TX DC differential mode low impedance
Lane-to-Lane Output Skew	$L_{TX-SKEW}$	-	-	1.25	ns	Static skew between any two transmitter lanes with a single link
AC Coupling Capacitor	C_{TX}	176	-	265	nF	All transmitters must be AC coupled. The AC coupling is either required within the media or within the transmitter component itself.
Crosslink Random Time-out	$T_{Crosslink}$	0	-	1	ms	This random time helps to resolve conflicts in crosslink configuration eventually resulting in only one downstream and one upstream port.

1) No test load is necessarily associated with this value.
2) Specified at the measurement point into a timing and voltage compliance test load and measured using the clock recovery function

Table 166 PCIe 4.0 Receiver Electrical and Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Unit Interval	UI	62.48125	62.5	62.51875	ps	Each UI is 62.5 ps +/-300 ppm. UI does not account for SSC dictated variations.[1)]
Minimum Receiver Eye Width	T_{RX-EYE}	0.30	-	-	UI	-.
Maximum Time Between the Jitter Median and Maximum Deviation form the Median	$T_{RX-MEDIAN-to-MAX-JITTER}$	0.30	-	-	UI	-
AC Peak Common Mode Input Voltage	$V_{RX-CM-ACp}$	-	-	75 for EH< 100 mVPP 125 for EH≥ 100 mVPP	mV	$V_{RX-CM-ACp} = \|V_{RX-D+} + V_{RX-D-}\|/2 – V_{RX-CM-DC})$ VRX-CM-DC = $DC_{(avg)}$ of $\|V_{RX-D+} + V_{RX-D-}\|/2$ See [2)]
Differential Return Loss	$RL_{RX-DIFF}$	-10	-	-	dB	Measured over 50 MHz to 1.25 GHz
Differential Return Loss	$RL_{RX-DIFF}$	-8	-	-6	dB	Measured over 1.25 GHz to 8 GHz
Common Mode Return Loss	RL_{RX-CM}	-6	-	-5	dB	Measured over 50 MHz to 8 GHz
Powered Down DC Input Impedance	$Z_{RX-HIGH-IMP-DC}$	≥10K (0-200 mV ≥20K (> 200 mV)	-	-	Ω	Required Rx D+ as well as D- DC impedance when the receiver terminations do not have power. See[2)]
Electrical Idle Detect Threshold	$V_{RX-IDLE-DET-DIFFp-p}$	65	-	175	mV	$V_{RX-IDLE-DET-DIFFp-p} = 2*\|V_{RX-D+} - V_{RX-D-}\|$
Unexpected Electrical Idle Enter Detect Threshold integration time	$V_{RX-IDLE-DET-DIFF-ENTERTIME}$		-	10	ms	An unexpected electrical idle ($V_{RX-DIFFp-p} < V_{RX-IDLE-DET-DIFFp-p}$) must be recognized no longer than $T_{RX-IDLE-DET-DIFF-ENTERTIME}$ to signal an unexpected idle condition.
Total Skew	$L_{RX-SKEW}$	-	-	5	ns	Skew across all lanes on a link. This includes variation in the length of a SKP ordered set (for example, COM and one to five SKP symbols) at the Rx as well as any delay differences arising from the interconnect itself.

1) No test load is necessarily associated with this value.

2) Impedance during all LTSSM states. When transitioning from a Fundamental Reset to Detect (the 30 initial state of the LTSSM) there is a 5 ms transition time before Receiver termination values must be met on all un-configured Lanes of a Port.

15.5.10 USB 2.0 Interface

These timings are according to USB Version 1.1/2.0 standard.

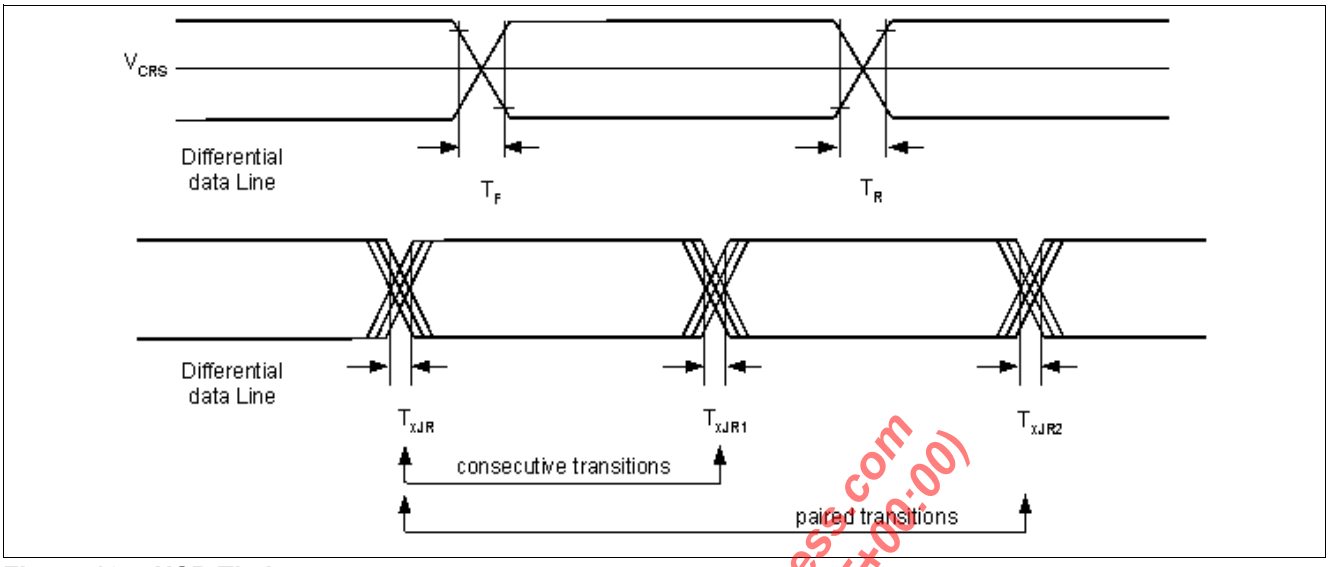


Figure 105 USB Timing

Table 167 USB 1.1 Port Timing Parameters

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Rise Time	T_R	4	–	20	ns	–
Fall Time	T_F	4	–	20	ns	–
Differential Rise and Fall Time Matching	T_{FRFM}	90	–	111.11	%	–
Driver Output Resistance	Z_{DRIV}	28	–	44	Ohm	–
Data Rate	f_D	11.97	–	12.03	Mbit/s	–
Frame Interval	T_{FRAME}	0.9995	–	1.0005	ms	–
Consecutive Frame Interval Jitter (without Clock Adjustment)	T_{RFI}	–	–	42	ns	–
Consecutive Frame Interval Jitter (with Clock Adjustment)	T_{RFIADJ}	–	–	126	ns	–
Total Output Jitter to Next Transition	T_{DJ1}	-3.5	–	3.5	ns	–
Total Output Jitter for Paired Transition	T_{DJ2}	-4	–	4	ns	–
Output Jitter for Differential Transition to SE0 Transition	T_{FDEOP}	-2	–	5	ns	–
Receive Jitter to Next Transition	T_{JR1}	-18.5	–	18.5	ns	–
Receive Jitter for Paired Transition	T_{JR2}	-9	–	9	ns	–
Output SEO Interval of EOP	T_{FEOPT}	160	–	175	ns	–
Input SEO Interval of EOP	T_{FEOPR}	82	–	–	ns	–
Width of SEO Interval during Differential Transition	T_{FST}	–	–	14	ns	–

Table 168 USB 2.0 High Speed Port Timing Parameters

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Rise Time	T_R	500	–		ps	–
Fall Time	T_F	500	–		ps	–
Driver Output Resistance	Z_{DRIV}	40.5	–	49.5	Ohm	–
Data Rate	f_D	479.760	–	480.240	Mbit/s	–
Frame Interval	T_{FRAME}	124.9375	–	125.0625	ms	–
Consecutive Frame Interval Jitter (without Clock Adjustment)	T_{RFI}	–	–	4	BT	–

15.5.11 USB 3.2 Gen 1

Table 169 USB 3.2 Gen 1 Tx Timing Parameters

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
UI (Unit Interval)	UI	199.94	–	200.06	ps	+/-300 ppm tolerance, excluding SSC
Differential Tx Voltage Swing	$T_{TX-DIFF-PP}$	0.8	1.0	1.2	V	–
Differential Tx Voltage Swing Low	$T_{TX-DIFF-PP-low}$	0.4	1.0	1.2	V	Low power mode Tx swing
Driver Output Resistance	$R_{TX-DIFF-Dc}$	72	–	120	Ohm	–
Transmitter DC Common Mode Resistance	R_{TX-DC}	18	–	30	Ohm	–
AC Coupling Capacitor	$C_{AC-COUPLING}$	75	–	200	nF	–
Maximum Slew Rate	$T_{CDR-CLEW}$	-	–	10	ms/s	–

Table 170 USB 3.2 Gen 1 Rx Timing Parameters

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
UI (Unit Interval)	UI	199.94	–	200.06	ps	+/-300 ppm tolerance, excluding SSC
LFPS Detect Threshold	$V_{RX-LFPS-DIFF-PP}$	100	-	300	mV	Below the minimum is considered noise.
Receiver Differential Resistance	$R_{RX-DIFF-DC}$	72	–	120	Ohm	–
Receiver DC Common Mode Resistance	R_{RX-DC}	18	–	30	Ohm	–

15.5.12 USB 3.2 Gen 2

Table 171 USB 3.2 Gen 2 Tx Timing Parameters

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
UI (Unit Interval)	UI	99.97	–	100.03	ps	+/-300 ppm tolerance, excluding SSC
Differential Tx Voltage Swing	$T_{TX\text{-}DIFF\text{-}PP}$	0.8	1.0	1.2	V	–
Differential Tx Voltage Swing Low	$T_{TX\text{-}DIFF\text{-}PP\text{-}low}$	0.4	1.0	1.2	V	Low power mode Tx swing
Driver Output Resistance	$R_{TX\text{-}DIFF\text{-}Dc}$	72	–	120	Ohm	–
Transmitter DC Common Mode Resistance	$R_{TX\text{-}DC}$	18	–	30	Ohm	–
AC Coupling Capacitor	$C_{AC\text{-}COUPLING}$	75	–	265	nF	–
SSC Rate	$SSC_{df/dt}$	-	–	1250	ppm/us	–

Table 172 USB 3.2 Gen 2 Rx Timing Parameters

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
UI (Unit Interval)	UI	99.97	–	100.03	ps	+/-300ppm tolerance, excluding SSC
LFPS Detect Threshold	$V_{RX\text{-}LFPS\text{-}DIFF\text{-}PP}$	100	-	300	mV	Below the minimum is considered noise.
Receiver Differential Resistance	$R_{RX\text{-}DIFF\text{-}DC}$	72	–	120	Ohm	–
Receiver DC Common Mode Resistance	$R_{RX\text{-}DC}$	18	–	30	Ohm	–

15.5.13 SATA 3.2 Interface

The various SATA standards are listed here:

- Gen1/2/3: 1.5/3/6 Gbps
- Gen1i/2i/3i: 1.5/3/6 Gbps with cable up to 1 m
- Gen1m/2m: 1.5/3 Gbps, cable up to 2 m
- Gen1u/2u/3u: 1.5/3/6 Gbps for Uhost, equivalent to 1 m cable plus mated connector pair

Refer to **[10]** for more information.

15.5.14 Serial Peripheral Interface (SPI)

Attention: The SPI internal latch clock delay is programmable via register SPI_CON:LTCLKDEL. The timing provided in this section is for the default register value of 0x00.

All 4 types of SPI operating modes are programmable by SPI control register CON. The 4 modes are:

Table 173 SPI Modes

SPI Mode	CON.PO	CON.PH
1	0	0
0	0	1
3	1	0
2	1	1

Figure 106 and Figure 107 show the SPI master and slave interface timing, respectively.

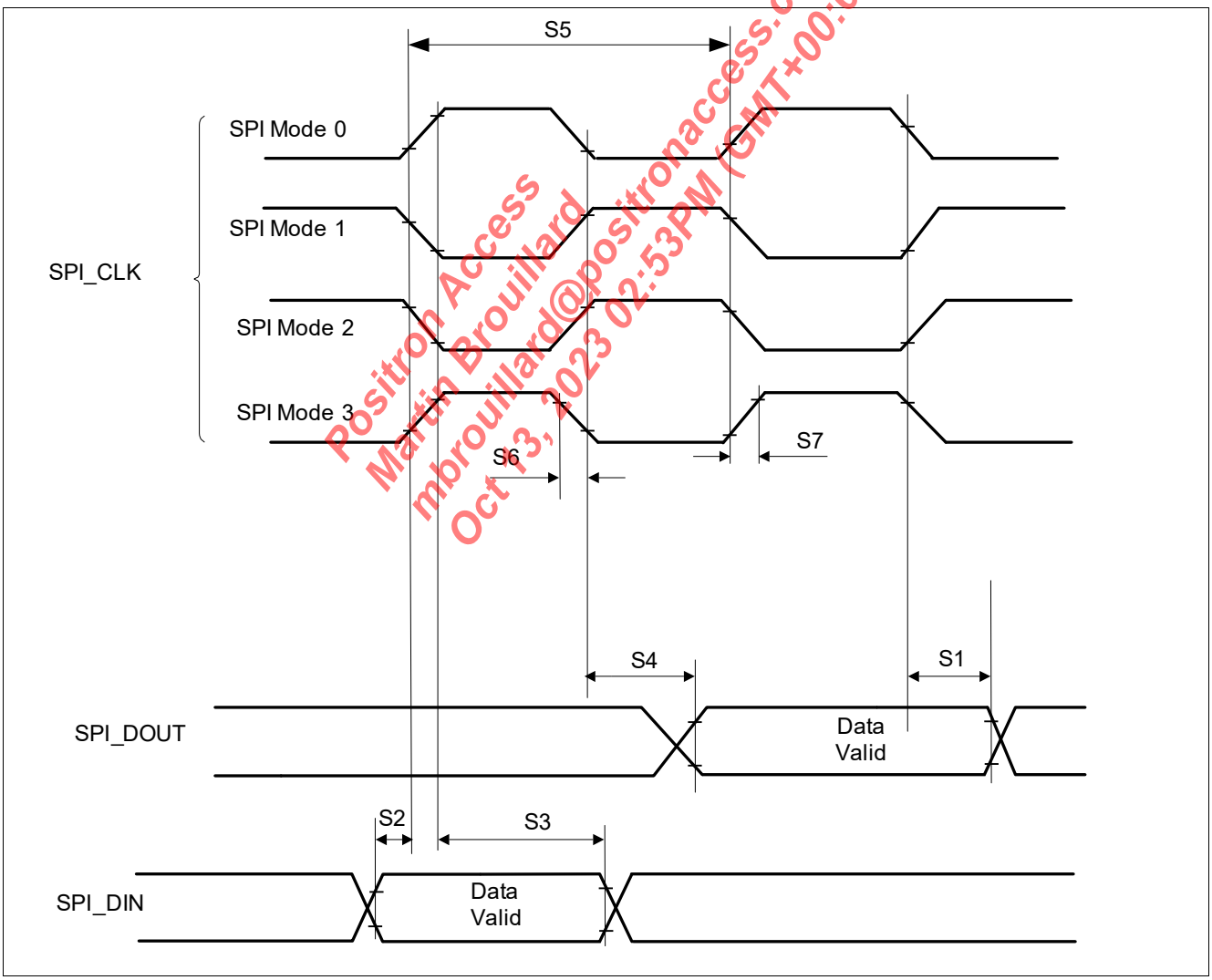


Figure 106 SPI Master Interface Timing

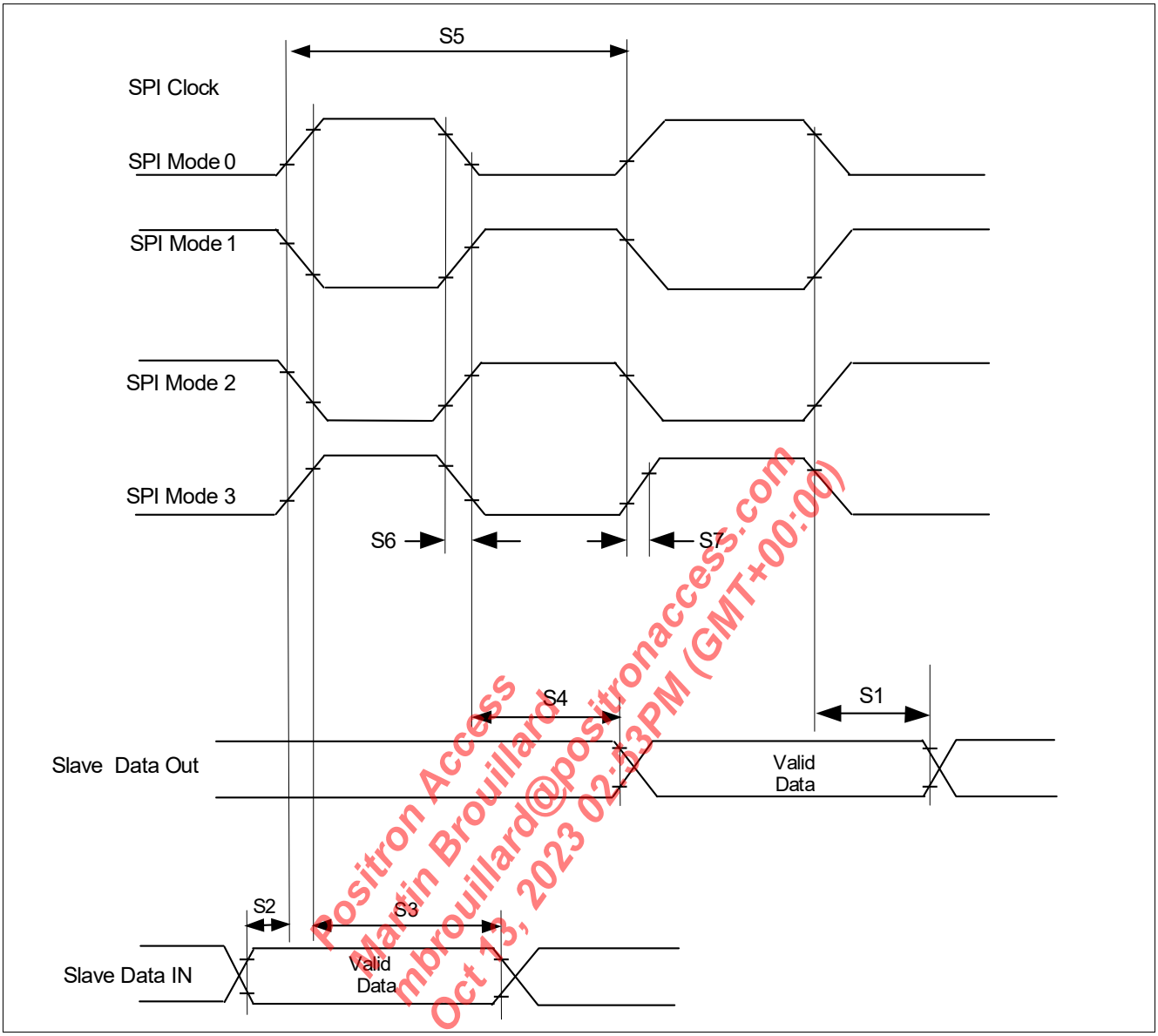


Figure 107 SPI Slave Timing

Table 174 SPI Interface Timing Parameters

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Master Mode						
TX Data Output Hold	S_1	S_5-S_4	–	–	ns	–
Tx Data Output Delay	S_4	0	–	2	ns	–
Rx Data Input Setup Time	S_2	2	–	–	ns	–
Rx Data Hold Time	S_3	0	–	–	ns	–
SPI Clock Period (Master Mode)	S_5	20	–	–	ns	–
SPI Clock Rising	S_7	0.1	–	–	V/ns	–
SPI Clock Falling	S_6	0.1	–	–	V/ns	–
Slave Mode						
Tx Data Output Hold	S_1	0	–	–	ns	–
Tx Data Output Delay	S_4	0	–	10	ns	–
Rx Data Input Setup Time	S_2	2	–	–	ns	–
Rx Data Hold Time	S_3	5	–	–	ns	–
SPI Clock Period (Slave Mode)	S_5	40	–	–	ns	–
SPI Clock Rising	S_7	0.1	–	–	V/ns	–
SPI Clock Falling	S_6	0.1	–	–	V/ns	–
SPI Clock Duty Cycle	–	48	–	52	%	–

15.5.15 Quad Serial Peripheral Interface (QSPI)

Attention: The SPI internal latch clock delay is programmable via register SPI_CON:LTCLKDEL. The timing provided in this section is for the default register value of 0x00.

All four types of SPI operating modes are programmable by the SPI control register CON. **Table 175** lists the four modes.

Table 175 QSPI Modes

SPI Mode	CON.PO	CON.PH
1	0	0
0	0	1
3	1	0
2	1	1

Figure 108 shows the QSPI master timing.

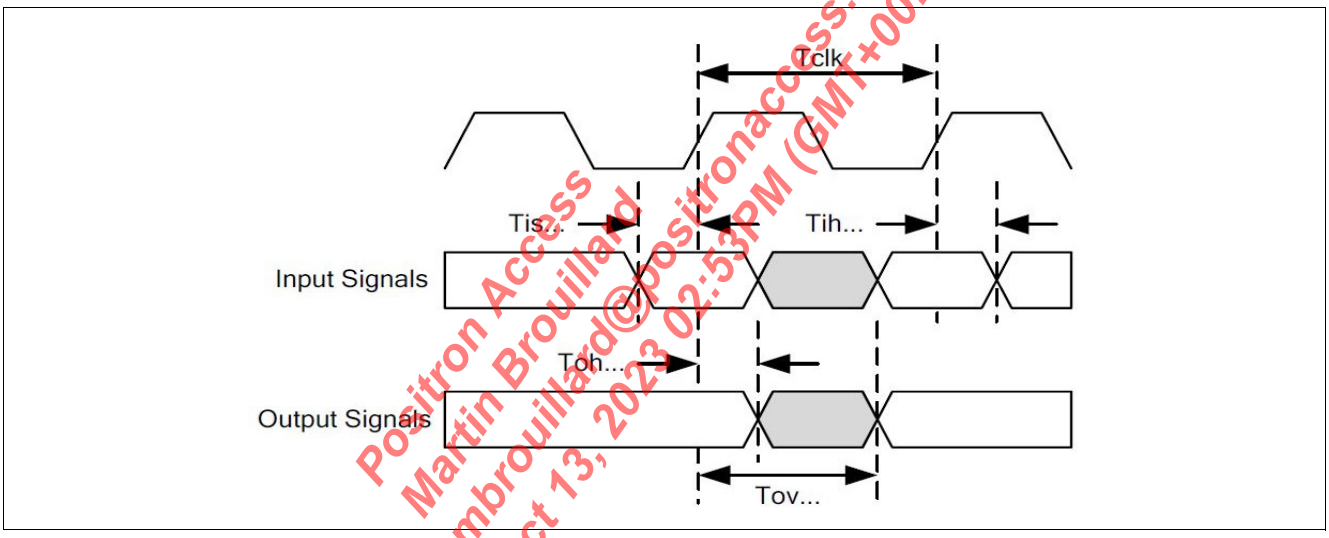


Figure 108 QSPI Master Interface Timing

Table 176 QSPI Interface Timing Parameters

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Master Mode						
TX Data Output Hold	*Toh*	5%	–	–	Tclk	for Tclk=5 ns for ref_clk of 200 MHz
Tx Data Output Delay	*Tov*	0	–	50%	Tclk	for Tclk=5 ns for ref_clk of 200 MHz
Rx Data Input Setup Time	Tis	-	–	35%	Tclk	for Tclk=5 ns for ref_clk of 200 MHz
Rx Data Hold Time	*Tih*	5%	–	–	Tclk	for Tclk=5 ns for ref_clk of 200 MHz
SPI Clock Period (Master Mode)	*Tclk*	10	–	–	ns	Target for 100 MHz

Table 176 QSPI Interface Timing Parameters (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
SPI Clock Rising	S_7	0.1	–	–	V/ns	–
SPI Clock Falling	S_6	0.1	–	–	V/ns	–

15.5.16 DDR3-SDRAM Timing

This section describes the supported DDR3 device timing.

15.5.16.1 Supported DDR-3 Device Timing

Table 177 describes the JEDEC standard, JESD79-3E.

Table 177 DDR3-SDRAM Device Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Clock Period	t_{ck666}	1.5	–	3.0	ns	DDR3-666 to DDR3-1333
Clock Period	t_{ck800}	1.25	–	3.0	ns	DDR3-1600 (to run at 800 MHz) except allowed SSC
Clock Period	t_{ck333}	3.0	3.0	3.0	ns	Power saving mode
CK High Level Width	t_{CH}	0.47	–	0.53	t_{ck}	Valid for all clock frequencies
CK Low Level Width	t_{CL}	0.47	–	0.53	t_{ck}	Valid for all clock frequencies
Clock Period Jitter	t_{JIT}	-80	–	80	*ps*	Valid for DDR3-1333
Clock Period Jitter	t_{JIT}	-70	–	70	*ps*	Valid for DDR3-1600 .
Write Command to DQS Associated Clock Edge	WL	RL - 1	–	–	t_{ck}	–
DQS Latching Rising Transitions to Associated Clock Edges	t_{DQSS}	-0.25	–	0.25	t_{ck}	Valid for DDR3-1333
DQS Latching Rising Transitions to Associated Clock Edges	t_{DQSS}	-0.27	–	0.27	t_{ck}	Valid for DDR3-1600
DQS Falling Edge to CK Setup Time	t_{DSS}	0.2	–	–	t_{ck}	Valid for DDR3-1333
DQS Falling Edge to CK Setup Time	t_{DSS}	0.18	–	–	t_{ck}	Valid for DDR3-1600
DQS Falling Edge Hold Time from CK	t_{DSH}	0.2	–	–	t_{ck}	Valid for DDR3-1333
DQS Falling Edge Hold Time from CK	t_{DSH}	0.18	–	–	t_{ck}	Valid for DDR3-1600
DQS, DQS#, Write Preamble	t_{WPRE}	0.9	–	–	t_{ck}	–
DQS, DQS#, Write Postamble	t_{WPST}	0.3	–	-	t_{ck}	–
Address and Control Input Setup Time AC 175 mV	t_{IS}	65	–	–	ps	Valid for DDR3-1333
Address and Control Input Setup Time AC 175 mV	t_{IS}	45	–	–	ps	Valid for DDR3-1600
Address and Control Input Setup Time AC 150 mV	t_{IS}	190	–	–	ps	Valid for DDR3-1333

Table 177 DDR3-SDRAM Device Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Address and Control Input Setup Time AC 150 mV	t_{IS}	170	–	–	ps	Valid for DDR3-1600
Address and Control Input Hold Time	t_{IH}	140	–	–	ps	Valid for DDR3-1333
Address and Control Input Hold Time	t_{IH}	120	–	–	ps	Valid for DDR3-1600
Control and Address Input Pulse Width for Each Input	t_{IPW}	0.62	–	-	ns	Valid for DDR3-1333
Control and Address Input Pulse Width for Each Input	t_{IPW}	0.56	–	-	ns	Valid for DDR3-1600
DQ and DM Input Setup Time AC 150 mV	t_{DS1}	0.03	–	–	ns	Valid for DDR3-1333
DQ and DM Input Setup Time AC 150 mV	t_{DS1}	0.01	–	–	ns	Valid for DDR3-1600
DQ and DM Input Hold Time	t_{DH1}	0.065	–	–	ns	Valid for DDR3-1333
DQ and DM Input Hold Time	t_{DH1}	0.045	–	–	ns	Valid for DDR3-1600
DQ and DM Input Pulse Width for Each Input	t_{DIPW}	0.40	–	–	t_{ck}	Valid for DDR3-1333
DQ and DM Input Pulse Width for Each Input	t_{DIPW}	0.36	–	–	t_{ck}	Valid for DDR3-1600
DQS Output Access Time from CK/$\overline{CK}$	t_{DQSCK}	-255	–	+255	t_{ck}	–
Data-out High Impedance Time from CK/$\overline{CK}$	t_{HZ}	–	–	250	ps	Valid for DDR3-1333
Data-out High Impedance Time from CK/$\overline{CK}$	t_{HZ}	x	–	225	ps	Valid for DDR3-1600
DQ Low Impedance Time from CK/$\overline{CK}$	$t_{LZ(DQ)}$	-500	–	250	ns	Valid for DDR3-1333
DQ Low Impedance Time from CK/$\overline{CK}$	$t_{LZ(DQ)}$	-450	–	225	ns	Valid for DDR3-1600
DQS, $\overline{DQS}$ Low Time	t_{DQSL}	0.45	–	0.55	t_{ck}	–
DQS, $\overline{DQS}$ High Time	t_{DQSH}	0.45	–	0.55	t_{ck}	–
Read Preamble	t_{RPRE}	0.9	–	-	t_{ck}	–
Read Postamble	t_{RPST}	0.3	–	-	t_{ck}	–
DQ/DQS Output Hold Time from DQS	t_{QH}	0.38	–	–	t_{ck}	–
DQ/DQS Skew (DQS and Associated DQ Signals)	t_{DQSQ}	–	–	125	ps	Valid for DDR3-1333
DQ/DQS Skew (DQS and Associated DQ Signals)	t_{DQSQ}	-	–	100	ps	Valid for DDR3-1600
Write Recovery Time	t_{WR}	15	–	–	ns	–

Table 177 DDR3-SDRAM Device Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
CKE Minimum Pulse Width (HIGH and LOW Pulse Width)	t_{CKE}	3	–	–	t_{ck}	–
ODT Turn-on Delay	t_{AOND}	WL-2	–	WL-2	t_{ck}	–
ODT Turn-on	t_{AON}	-250	–	250	ns	Valid for DDR3-1333
ODT Turn-on	t_{AON}	-225	–	225	ns	Valid for DDR3-1600
ODT Turn-on (Power-Down Mode)	t_{AON}	2	–	8.5	ns	–
ODT Turn-off Delay	t_{AOFD}	WL-2	–	WL-2	t_{ck}	–
ODT Turn-off	t_{AOF}	0.3	–	0.7	t_{ck}	–
Mode Register Set Command Cycle Time	t_{MRD}	4	–	–	t_{ck}	–
MRS Command to ODT Update Delay	t_{MOD}	12	–	-	t_{ck}	–
CL Related Parameters - DDR3-1333						
Row to Column Command Time	t_{rcd}	10.5	–	15	ns	CL-nRCD-nRP 7-7-7 to 10-10-10
$\overline{CAS}$ to $\overline{CAS}$ Command Delay	t_{CCD}	4	–	–	t_{ck}	-
Row Active Time	t_{ras}	36	–	$9*t_{REFI}$	*ns*	CL-nRCD-nRP 7-7-7 to 10-10-10
RAS Cycle Time	t_{rc}	46.5	–	51	ns	CL-nRCD-nRP 7-7-7 to 10-10-10
Precharge Time	t_{rp}	10.5	–	15	ns	CL-nRCD-nRP 7-7-7 to 10-10-10
CL Related Parameters - DDR3-1600						
Row to Column Command Time	t_{rcd}	10	–	13.75	ns	CL-nRCD-nRP 8-8-8 to 11-11-11
$\overline{CAS}$ to $\overline{CAS}$ Command Delay	t_{CCD}	4	–	–	t_{ck}	-
Row Active Time	t_{ras}	35	–	9*tREFI	*ns*	CL-nRCD-nRP 8-8-8 to 11-11-11
RAS Cycle Time	t_{rc}	45.25	–	48.75	ns	CL-nRCD-nRP 8-8-8 to 11-11-11
Precharge Time	t_{rp}	10	–	13.75	ns	CL-nRCD-nRP 8-8-8 to 11-11-11

15.5.17 DDR4-SDRAM Timing

This section describes the supported DDR4 device timing.

15.5.17.1 Supported DDR-4 Device Timing

Table 177 describes the JEDEC standard, JESD79-4.

Table 178 DDR4-SDRAM Device Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Clock Period	t_{ck800}	-	1.25	3.0	ns	DDR4-1600
Clock Period	t_{ck1333}	-	0.75	3.0	ns	DDR4-2667 (to run at 1333 MHz) except allowed SSC
Clock Period	t_{ck1600}	-	0.625	3.0	ns	DDR4-3200 (to run at 1600 MHz) except allowed SSC
Clock Period	t_{ck333}	3.0	3.0	3.0	ns	Power Saving mode
CK High Level Width	t_{CH}	0.48	–	0.52	t_{ck}	Valid for all clock frequencies
CK Low Level Width	t_{CL}	0.48	–	0.52	t_{ck}	Valid for all clock frequencies
Clock Period Jitter	t_{JIT}	-	–	75	*ps*	Valid for DDR4-2667
Clock Period Jitter	t_{JIT}	-	–	62.5	*ps*	Valid for DDR4-3200
Write Command to DQS Associated Clock Edge	WL	RL - 1	–	–	t_{ck}	
DQS Latching Rising Transitions to Associated Clock Edges	t_{DQSS}	-0.27	–	0.27	t_{ck}	
DQS Falling Edge to CK Setup Time	t_{DSS}	0.18	–	-	t_{ck}	
DQS Falling Edge Hold Time from CK	t_{DSH}	0.18	–	-	t_{ck}	
DQS, DQS#, Write Preamble	t_{WPRE}	0.9	–	–	t_{ck}	
DQS, DQS#, Write Postamble	t_{WPST}	0.3	–	-	t_{ck}	
Address and Control Input Setup Time	t_{IS}	55	–	–	ps	
Address and Control Input Setup Time VREF	t_{ISVREF}	145	–	–	ps	
Address and Control Input Hold Time	t_{IH}	80	–	–	ps	
Control and Address Input Pulse Width for Each Input	t_{IPW}	385	–	-	ps	
Control and Address Input Pulse Width for Each Input	t_{IPW}	0.385	–	-	ns	
DQ and DM Input Setup Time A	t_{DS1}	0.03	–	–	ns	

Table 178 DDR4-SDRAM Device Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
DQ and DM Input Hold Time	t_{DH1}	0.065	–	–	ns	
DQ and DM Input Pulse Width for Each Input	t_{DIPW}	0.58	–	–	t_{ck}	
DQ and DM Input Pulse Width for Each Input	t_{DIPW}	0.36	–	–	t_{ck}	
DQS Output Access Time from CK/$\overline{CK}$	t_{DQSCK}	-170	170	170	ps	
Data-out High Impedance Time from CK/$\overline{CK}$	t_{HZ}	–	–	140	ps	
Data-out High Impedance Time from CK/$\overline{CK}$	t_{HZ}	x	–	140	ps	
DQ Low Impedance Time from CK/$\overline{CK}$	$t_{LZ(DQ)}$	-340	–	140	ps	
DQS, $\overline{DQS}$ Low Time	t_{DQSL}	0.4	–	-	t_{ck}	–
DQS, $\overline{DQS}$ High Time	t_{DQSH}	0.4	–	-	t_{ck}	–
Read Preamble	t_{RPRE}	0.9	–	-	t_{ck}	–
Read Postamble	t_{RPST}	0.33	–	-	t_{ck}	–
DQ/DQS Output Hold Time from DQS	t_{QH}	tbd	–	–	t_{ck}	–
DQ/DQS Skew (DQS and Associated DQ Signals)	t_{DQSQ}	–	–	tbd	ps	-
Write Recovery Time	t_{WR}	15	–	–	ns	–
CKE Minimum Pulse Width (High and Low Pulse Width)	t_{CKE}	3	–	–	t_{ck}	–
ODT Turn-on Delay	t_{AOND}	WL-2	–	–	ns	–
ODT Turn-on (Power-Down Mode)	t_{AONAS}	1	–	9	ns	–
ODT Turn-off Delay	t_{AOFD}	WL-2	–	–	ns	–
ODT Turn-off	t_{AOFAS}	1	–	9	ns	–
Mode Register Set Command Cycle Time	t_{MRD}	9	–	–	t_{ck}	–
MRS Command to ODT Update Delay	t_{MOD}	max(24tck,15ns)	–	–	t_{ck}	–
CL Related Parameters - DDR4-2133						
Row to Column Command Time	t_{rcd}	14.06	–	–	ns	CL-nRCD-nRP 15-15-15
$\overline{CAS}$ to $\overline{CAS}$ Command Delay, Different Bank Group	t_{CCD_S}	4	–	–	t_{ck}	-
$\overline{CAS}$ to $\overline{CAS}$ Command Delay, Same Bank Group	t_{CCD_L}	5	–	–	t_{ck}	-

Table 178 DDR4-SDRAM Device Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Row Active Time	t_{ras}	33	–	9*t_{REFI}	*ns*	CL-nRCD-nRP 15-15-15
RAS Cycle Time	t_{rc}	47.06	–	$t_{ras\ plus}$ t_{rp}	ns	CL-nRCD-nRP 15-15-15
Precharge Time	t_{rp}	14.06	–	-	ns	CL-nRCD-nRP 15-15-15
CL Related Parameters - DDR4-2667						
Row to Column Command Time	t_{rcd}	13.5	–	-	ns	CL-nRCD-nRP 18-18-18
$\overline{CAS}$ to $\overline{CAS}$ Command Delay	t_{CCD}	4	–	–	t_{ck}	-
Row Active Time	t_{ras}	32	–	9*tREFI	*ns*	CL-nRCD-nRP 18-18-18
RAS Cycle Time	t_{rc}	45.5	–	$t_{ras\ plus}$ t_{rp}	ns	CL-nRCD-nRP 18-18-18
Precharge Time	t_{rp}	13.5	–	-	ns	CL-nRCD-nRP 18-18-18
CL Related Parameters - DDR4-3200						
Row to Column Command Time	t_{rcd}	13.75	–	–	ns	CL-nRCD-nRP 22-22-22
$\overline{CAS}$ to $\overline{CAS}$ Command Delay	t_{CCD}	4	–	–	t_{ck}	–
Row Active Time	t_{ras}	32	–	9*tREFI	*ns*	CL-nRCD-nRP 22-22-22
RAS Cycle Time	t_{rc}	45.75	–	$t_{ras\ plus}$ t_{rp}	ns	CL-nRCD-nRP 22-22-22
Precharge Time	t_{rp}	13.75	–	–	ns	CL-nRCD-nRP 22-22-22

15.5.18 LPDDR4-SDRAM Timing

This section describes the supported LPDDR4 device timing.

15.5.18.1 Supported LPDDR-4 Device Timing

Table 177 describes the JEDEC standard, JESD209-4B.

Table 179 LPDDR4-SDRAM Device Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Clock Period	t_{ck1600}	–	0.625	3.0	ns	LPDDR4-3200
Clock Period	t_{ck1866}	–	0.468	3.0	ns	LPDDR4-3773 (to run at 933 MHz) except allowed SSC
Clock Period	t_{ck333}	3.0	3.0	3.0	ns	Power saving mode
CK High Level Width	t_{CH}	0.48	–	0.52	t_{ck}	Valid for all clock frequencies
CK Low Level Width	t_{CL}	0.48	–	0.52	t_{ck}	Valid for all clock frequencies
Clock Period Jitter	t_{JIT}	–	–	40	*ps*	Valid for LPDDR4-3200
Write Command to DQS Associated Clock Edge	WL	RL - 1	–	–	t_{ck}	
DQS Latching Rising Transitions to Associated Clock Edges	t_{DQSS}	0.75	–	1.25	t_{ck}	
DQS Falling Edge to CK Setup Time	t_{DSS}	0.2	–	–	t_{ck}	
DQS Falling Edge Hold Time from CK	t_{DSH}	0.2	–	–	t_{ck}	
DQS, DQS#, Write Preamble	t_{WPRE}	1.8	–	–	t_{ck}	
DQS, DQS#, Write Postamble	t_{WPST}	0.4	–	–	t_{ck}	0.5 tCK write postamble
DQS Output Low Impedance Time from CK/$\overline{CK}$	$t_{LZ(DQS)}$	tbd	–	–	ps	
DQS-out High Impedance Time from CK/$\overline{CK}$	$t_{HZ(DQS)}$	–	–	tbd	ps	
Data-out High Impedance Time from CK/$\overline{CK}$	$t_{HZ(DQ)}$	x	–	tbd	ps	
DQ Low Impedance Time from CK/$\overline{CK}$	$t_{LZ(DQ)}$	tbd	–	–	ps	
DQS, $\overline{DQS}$ Low Time	t_{DQSL}	tCL-0.05	–	–	t_{ck}	–
DQS, $\overline{DQS}$ High Time	t_{DQSH}	tCL-0.05	–	–	t_{ck}	–
Read Preamble	t_{RPRE}	1.8	–	–	t_{ck}	–
Read Postamble	t_{RPST}	0.4	–	–	t_{ck}	For 0.5 tCK READ postamble

Table 179 LPDDR4-SDRAM Device Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
DQ/DQS Output Hold Time from DQS	t_{QH}	tbd	–	–	t_{ck}	–
DQ/DQS Skew (DQS and Associated DQ Signals)	t_{DQSQ}	–	–	0.18	UI	–
Write Recovery Time	t_{WR}	maximum (18ns, 6nCK)	–	–	ns	–
CKE Minimum Pulse Width (High and Low Pulse Width)	t_{CKE}	3	–	–	t_{ck}	–
Mode Register Set Command Cycle Time	t_{MRD}	maximum (14ns, 10nCK)	–	–	t_{ck}	–
MRS Command to ODT Update Delay	t_{MOD}	maximum (24tck, 15ns)	–	–	t_{ck}	–
CL Related Parameters						
Row to Column Command Time	t_{rcd}	maximum (18 ns, 4nCK)	–	–	ns	–
$\overline{CAS}$ to $\overline{CAS}$ Command Delay, Different Bank Group	t_{CCD_S}	8	–	–	t_{ck}	–
$\overline{CAS}$ to $\overline{CAS}$ Command Delay, Same Bank Group	t_{CCD_L}	8	–	–	t_{ck}	–
Row Active Time	t_{ras}	maximum (42ns, 3nCK)	–	$9*t_{REFI}$	*ns*	–
RAS Cycle Time	t_{rc}	tRAS+tRPpb/tRPab	–	$t_{ras\ plus}\ t_{rp}$	ns	–
Precharge Time	t_{rp}	maximum (18ns, 4nCK)	–	–	ns	Single Bank
Four-bank ACTIVAT Time	t_{FAW}	40			ns	

15.5.19 Serial Shift-Out Register Interface

Figure 109 shows the Serial Shift Register Interface timing.

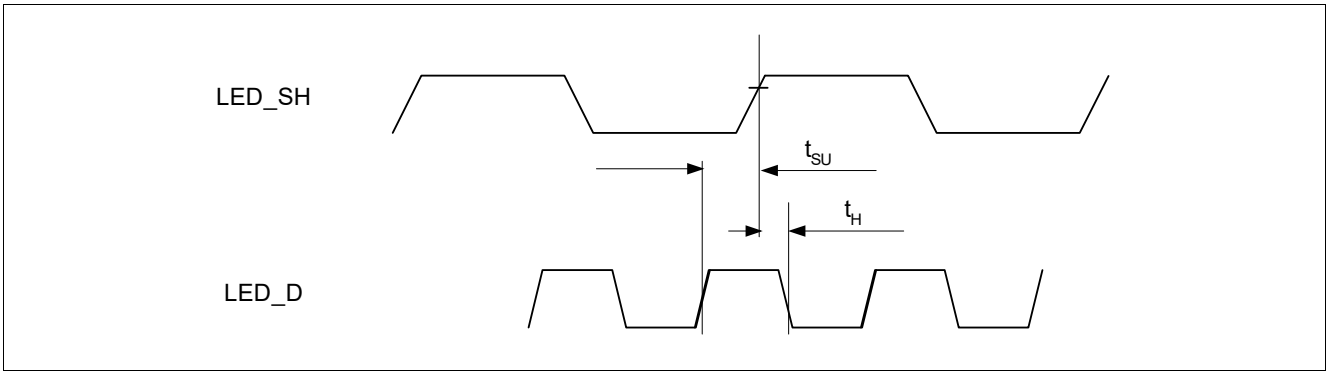


Figure 109 Serial Shift Register Timing

Table 180 describes the timing values.

Table 180 Shift Data Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Setup Time Data to Shift Clock	t_{SU}	15	–	–	ns	–
Hold Time Data to Shift Clock	t_H	5	–	–	ns	–

15.5.20 I2C Interface

Figure 110 shows the I2C interface timing.

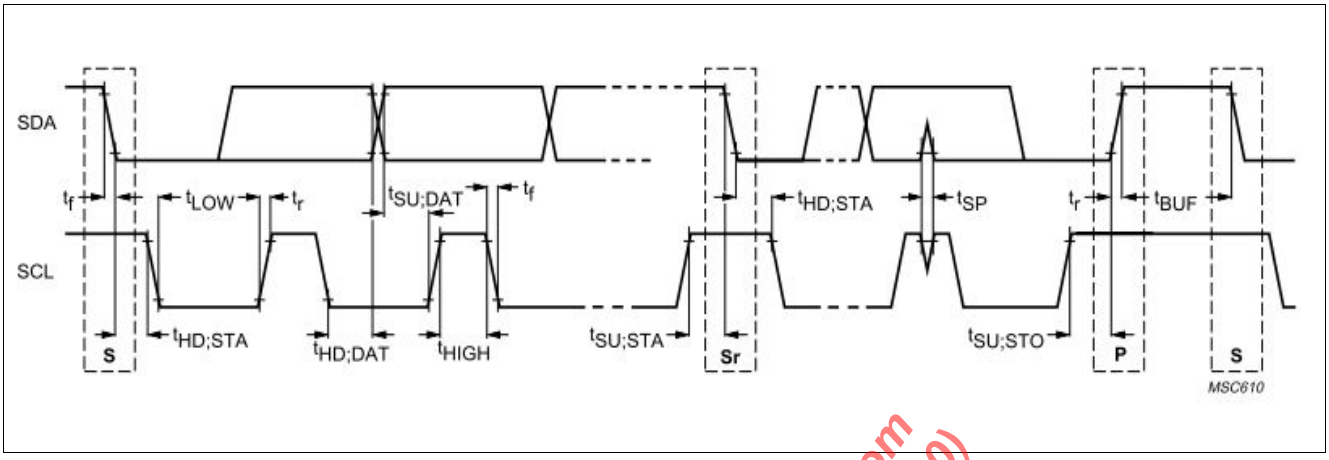


Figure 110 I2C Timing

Table 181 describes the timing values.

Table 181 I2C Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
SCL Frequency	f_{SCL}	0	–	100	kHz	–
Setup Time Data to Shift Clock	$t_{SU,DAT}$	250	–	–	ns	–
Hold Time Data to Shift Clock	$t_{HD,DAT}$	0	–	3.45	us	–
Setup Time START to Shift Clock	$t_{SU,STA}$	4700	–	–	ns	–
Hold Time START, STOP to Shift Clock	$t_{HD,STA/STO}$	4.0	–	-	us	–
Low Time	t_{LOW}	4700	–	–	ns	–
High Time	t_{HIGH}	4000	–	–	ns	–
Rising Time	t_r	–	–	1000	ns	–
Falling Time	t_f	–	–	300	ns	–
Bus Free Time	t_{BUF}	4700	–	–	ns	–

15.5.21 I2S Interface

Figure 110 shows the I2S interface timing.

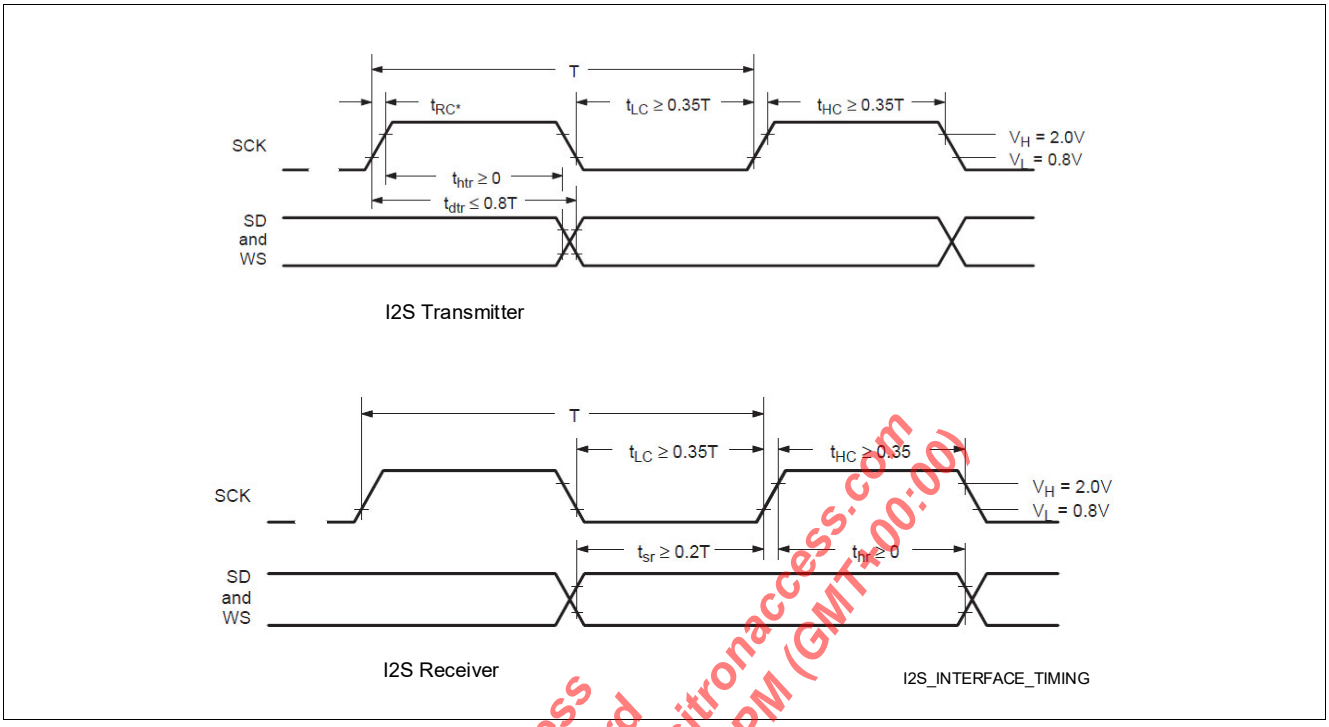


Figure 111 I2S Timing

Table 181 describes the timing values.

Table 182 I2S Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
SCK Period	T_{tr} or T_t	122	–	7353	ns	Bit clock 136 kHz to 8.192 MHz
Receiver Setup Time Data/WS to Shift Clock	t_{Sr}	0.2Ttr	–	–	ns	–
Receiver Hold Time Data/WS to Shift Clock	t_{Hr}	0	–	-	ns	–
Transmitter TX Delay	t_{dtr}	-	–	0.8Ttr	ns	–
Transmitter Hold Time	t_{Htr}	0	–	-	ns	–
SCK Low Time	t_{LC}	0.35Ttr	–	–	ns	–
SCK High Time	t_{HC}	0.35Ttr	–	–	ns	–
Rising Time	t_{RC}	–	–	0.15Ttr	ns	–

15.5.22 SDIO Card Interface

Figure 112 shows the SDIO interface timings.

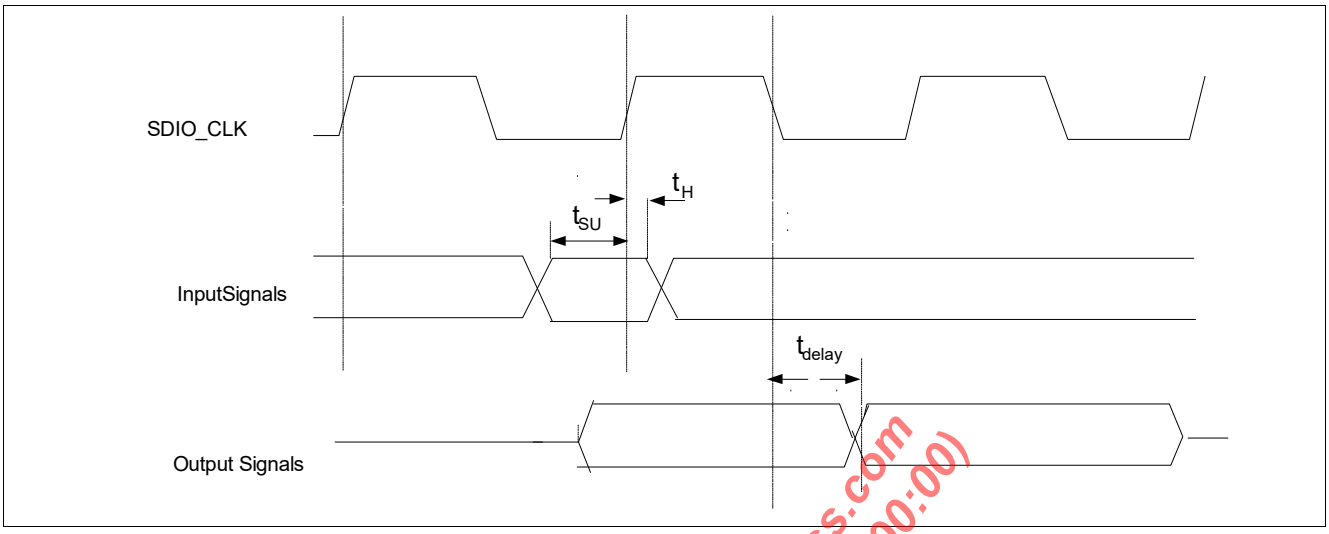


Figure 112 SDIO Timing Diagram - 25 MHz Mode

Table 183 SDIO (3.3 V) 25 and 50 MHz Mode Interface Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Clock Period		40	–	–	ns	25 MHz
		20	–	–	ns	50 MHz
Data Input Setup Time	t_{SU}	5	–	–	ns	Inputs: CMD, DAT
Data Input Hold	t_H	5	–	–	ns	–
Data Output Delay	t_{delay1}	2	–	14	ns	Output delay during data transfer mode
Data Output Delay	t_{delay2}	2	–	50	ns	Output delay during identification mode

Table 184 SDIO (1.8 V) SDR25/50/104 Mode Interface Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Clock Period	tck	20	–	–	ns	SDR25 MHz
		10	–	–	ns	SDR50 MHz
		4.8	–	–	ns	SDR104 mode, may not be supported.
Clock Rising/Falling Time				0.2 tck	ns	2 ns at 100 MHz for 10 pF load
Data Input Setup Time	t_{SU}	3	–	–	ns	Inputs: CMD, DAT
Data Input Hold	t_H	0.8	–	–	ns	–
Data Output Delay	t_{delay1}	1.5	–	14	ns	Output delay during SDR25 mode
Data Output Delay	t_{delay2}	1.5	–	7.5	ns	Output delay during SDR50 mode

15.5.23 eMMC 5.1 Interface

Figure 112 shows the eMMC 5.1 interface timings, which are according to JESD84-B50.

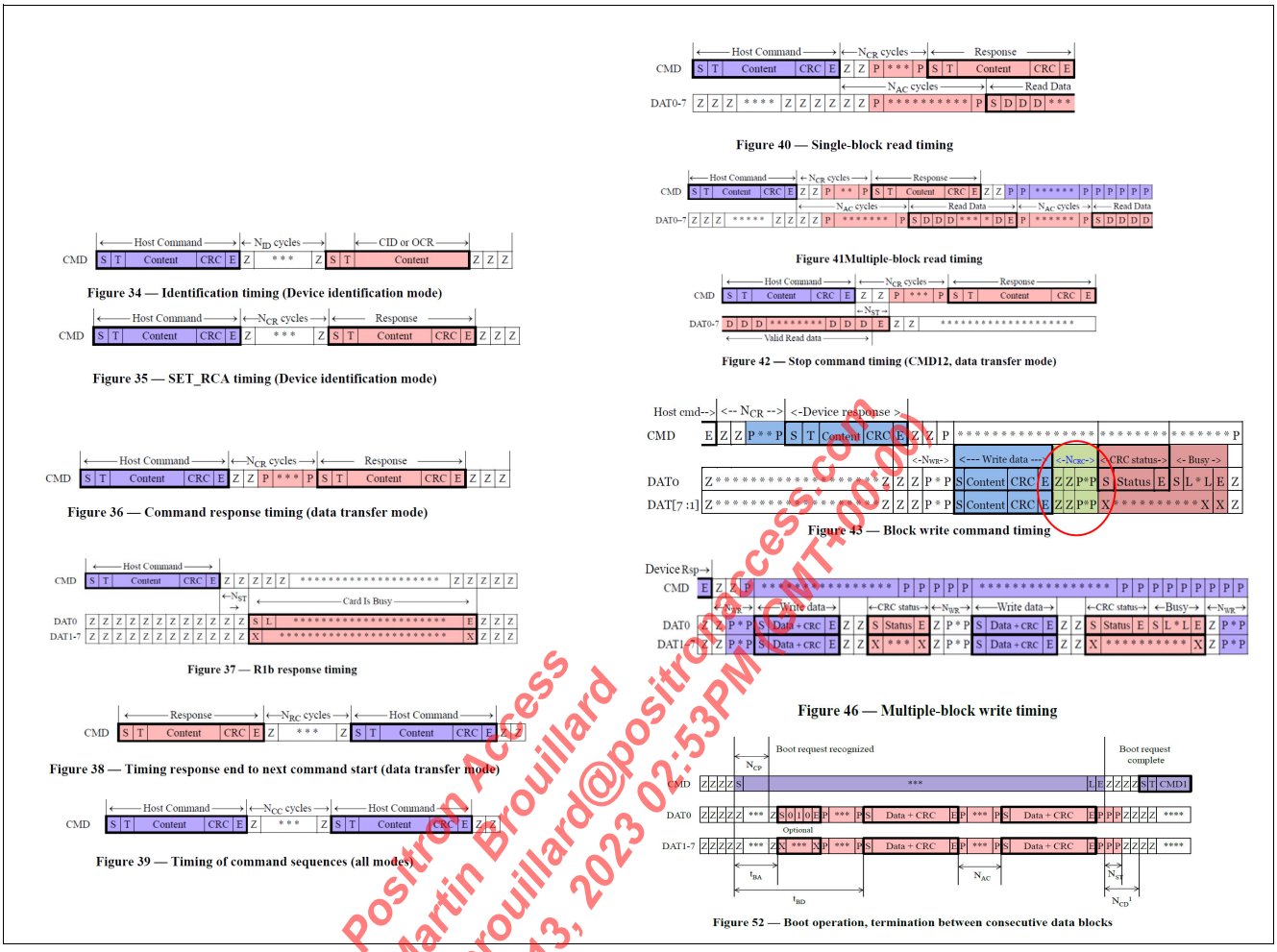


Figure 113 eMMC Timing Diagrams

Table 185 eMMC Interface Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Address to Command	N_{AC}	2	–	10*(TAAC*FOP+100*NSAC)	clock cycles	example: for FOP =10 MHz, the NAC max=150K cycles.
Command to Command Delay	N_{CC}	8	–	–	clock cycles	–
Command to Data	N_{CD}	56	–	–	clock cycles	–
Boot Command	N_{CP}	74	–	-	clock cycles	–
Command to Response	N_{CR}	2	–	64	clock cycles	–

Table 185 eMMC Interface Timing (cont'd)

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
ID Return Delay	N_{ID}	5	–	5	clock cycles	–
Response to Command	N_{RC}	8	–	-	clock cycles	–
Busy	N_{ST}	2	–	–	clock cycles	–
Write Timing	N_{WR}	2	–	-	clock cycles	–
STOP Command to Program Delay	N_{SB-A_B}	4	–	4	clock cycles	–
Bus Mode Change	N_{SC}	8	–	-	clock cycles	–
Boot Timing 1	t_{BD}	-	–	1	s	–
Boot Timing 2	t_{BA}	-	–	50	ms	–

15.5.24 Test Interface

The test interface is used to debug the CPU.

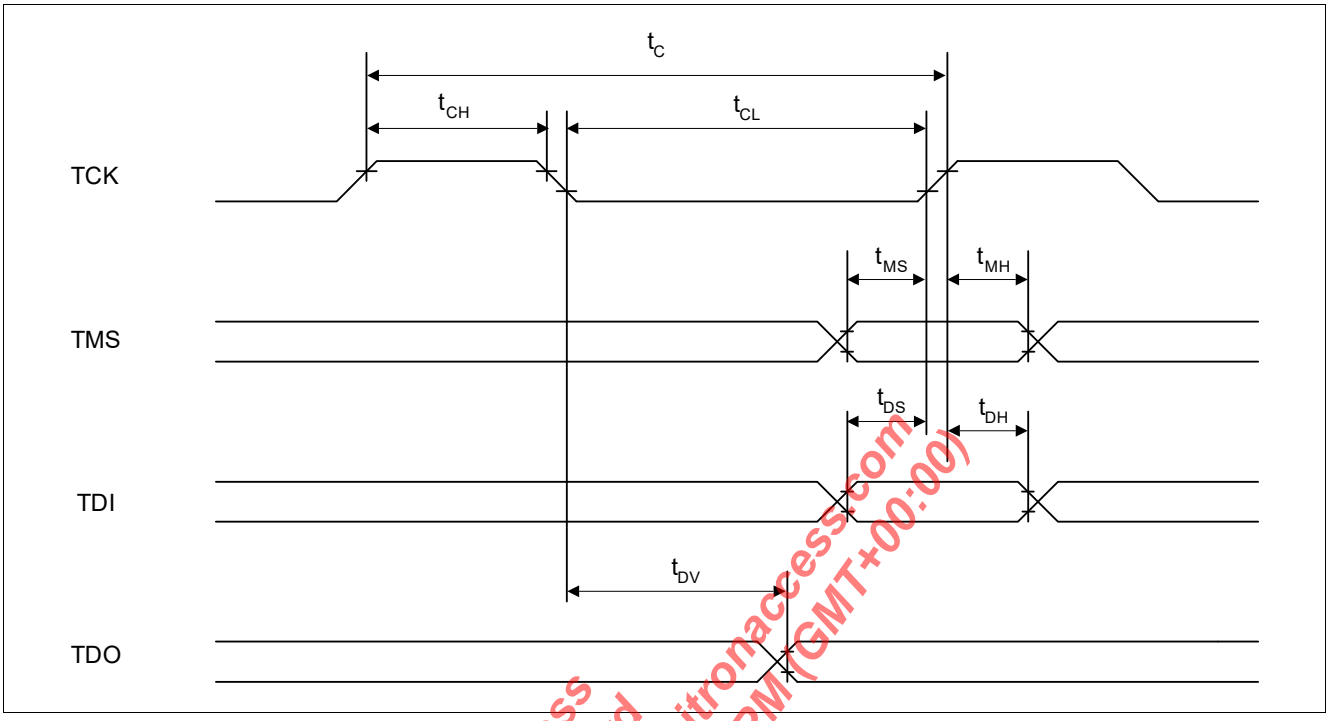


Figure 114 Test Interface Timing

Table 186 and Table 187 describe the timing values.

Table 186 Test Interface Clock

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
TCK Clock Period	t_C	100	–	–	ns	–
TCK High Time	t_{CH}	40	–	–	ns	–
TCK Low Time	t_{CL}	40	–	–	ns	–

Table 187 JTAG Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
TMS Setup Time	t_{MS}	40	–	–	ns	–
TMS Hold Time	t_{MH}	40	–	–	ns	–
TDI Setup Time	t_{DS}	40	–	–	ns	–
TDI Hold Time	t_{DH}	40	–	–	ns	–
Hold: $\overline{\text{TRST}}$ after TCK	t_{HD}	10	–	–	ns	–
TDO Valid Delay	t_{DV}	–	–	100	ns	–

15.5.25 Clock Supply Specification

In the reference design, the crystal is attached to the SoC and must follow the specification in **Table 188**. For direct oscillator attached, the specification is as in **Table 189**.

Table 188 Specification of the 40 MHz Crystal

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Nominal Frequency	–	–	40	–	MHz	Oscillation mode is fundamental.
Frequency Tolerance	–	-30	–	+30	ppm	At 25°C
Frequency Stability	–	-30	–	+30	ppm	Within working temperature range; reference to 25°C frequency
Series Resonant Resistance	–	–	–	40	Ω	–
Drive Level	–	0.08	0.10	0.2	mW	–
Load Capacitance	C_L	16	-	30	pF	–
Shunt Capacitance	C_0	–	-	7	pF	–
XTAL1 Voltage Input Level	-	0.5	-	0.8	V	

Table 189 Specification of the 40 MHz Oscillator

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Nominal Frequency	–	–	40	–	MHz	Oscillation mode is fundamental.
Frequency Tolerance	–	-30	–	+30	ppm	At 25°C
Frequency Stability	–	-30	–	+30	ppm	Within working temperature range; reference to 25°C frequency
XTAL1 Voltage Input Level	-	0.05	-	0.8	V	Single ended input
XTAL1/2 Voltage Input Level	-	0.1	-	0.8	V	Differential ended input
Long-term Jitter	-	-	1.0	1.5	ps	10 kHz -10 MHz

Terminology

A

ACA	AnyWan™ Common Architecture
AS	Asset Store
ADP	Accelerated Data Path
AES	Advanced Encryption Standard
ANSI	American National Standards Institute

B

BG	Baudrate Timer/Reload Register
BL	Boot Loader
BM	Buffer Manager
BRH	Boot ROM Header

C

CM	Crypto Module
CML	Common-Mode Logic
CPS	Coherent Processing Subsystem
CQE	Carrier Grade QoS Engine
CQHCI	Command Queue Host Controller Interface
CQM	Central Queue Manager

D

DBI	Data Bus inversion
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard

E

EBU	External Bus Unit
ECC	Error Correction Code
ECDSA	Elliptic Curve Digital Signature Algorithm
EOP	End of Packet
EPU	Energy Process Unit

F

FDV	Fractional Divider Register
FMT	Fixed Memory Translation
FSB	First Stage Bootloader
FSQM	Free Segment Queue Manager

G

GPIO	General Purpose Input/Output
GSWIP	Gigabit Switch IP
GSWIP-O	Gigabit Switch IP-OMCI

H

HSIO	High-Speed Input/Output
HSTL	High-Speed Transceiver Logic
HUK	Hardware Unique Key

I

IA	Intel Architecture
IED	In-line Encryption Device

J

JTAG	Joined Test Action Group

K

KBKDF	Key-based Key Derivation Function

L

LAN	Local Area Network
LRO	Large Reassembly Offloading

M

MPS	Multi-Processor System

N

NPU	Network Processing Unit
NoC or NOC	Network on Chip
NVM	Non-Volatile Memory

O

OPC	Operating Point Controller
OS	Operating System
OTP	One Time Programmable Memory

P

PAE	Packet Acceleration Engine
PCB	Printed Circuit Board
PCS	Physical Coding Sublayer
PDC	Power Domain Controller
PMIC	Power Management Integrated Circuit
PON	Passive Optical Network
POR	Power-on Reset
PPM	Protected Platform Mode
PPS	Pulse Per Second

Q

QoS	Quality of Service

R

RoT	Root of Trust
Rx	Receive

S

SB	Secure Boot
SBCR	Secure Boot Confidentiality Root
SBE	Storage Boost Elements
SHA	Secure Hash Standard
SJLI	Secure Jump Load Instruction
SMP	Symmetric Multi-Processing
SoC	System on Chip
SPE	Secure Platform Engine
SPI	Serial Peripheral Interface
SSB	Shared SRAM Buffer
SUSI	Signing Utilities for Signing Images
SyncE	Synchronous Ethernet

T

TC	Thread Context
TEE	Trusted Execution Environment
TEP	Trusted Execution Processor
TPI	Twisted Pair Interface
TRNG	True Random Number Generator
TSO	Transmit Segmentation Offload
Tx	Transmit

U

UART	Universal Asynchronous Receiver/Transmitter

V

VM	Virtual Machine
VPI	Virtual Processor Engine

W

WLAN	Wireless Local Area Network

Literature References

[1] AnyWAN™ Development Kit URX851 V2.2.1 HDK HW7.10 Hardware Design Guide Rev. 1.0

[2] AnyWAN™ Development Kit URX851 V2.3.1 HDK HW8.11 Release Notes Rev. 1.0

[3] Universal Gateway Software 8.x Software Overview Rev. 2.0

[4] Intel 64 and IA-32 Architectures Software Developer Manuals

[5] AnyWAN™ Broadband SoCs URX85x/MxL256xx Power Management Application Note Rev. 1.2

[6] AnyWAN™ SoC URX851/URX850/MxL25641 Errata Sheet Rev. 1.5

[7] Vishay TEMT6000 Ambient Light Sensor Data Sheet
https://www.vishay.com/docs/81579/temt6000.pdf

Attention: Refer to the latest revisions of the documents.

Standard References

[8] Serial-GMII Specification: Revision 1.8, Cisco Systems, April 2005
https://archive.org/details/sgmii

[9] UM10204 I^2C-bus Specification and User Manual Rev.6, April 2014

[10] Serial ATA (SATA) Specifications
https://sata-io.org/

Attention: Refer to the latest revisions of the documents.