LAN9645xF

5/7/9-Port Gigabit Ethernet Switch with TSN, AVB, and Redundancy Managed Modes

Highlights

- 5, 7, or 9-Port Gigabit Ethernet Switch with optional Managed mode
- 5x integrated 10/100/1000BASE-T PHYs with LED driver outputs
- Support for 2x RMII/RGMII, 1x SGMII/SGMII+/ Quad-SGMII, 1x SGMII/SGMII+
- Support for external PHY and SFP modules
- Flexible configuration options
  - Pin strapping and EEPROM for unmanaged operation
  - Linux DSA with SPI, I^2C, MIIM, and in-band (VRAP) interface options for managed operation
- Support for Audio Video Bridging (AVB)
- Support for Time-Sensitive Networking (TSN)
- Hardware support for Industrial MRP, PRP DANP, HSR DANH, RSTP, and unmanaged FRER
- 128-pin TQFP (14 x 14 mm) or 156-pin VQFN-DR (12 x 12 mm) package
- Industrial temperature range support (-40°C to +85°C)*

Target Applications

- Industrial control and Process Automation
- Networked test and measurement equipment
- Telematics & Smart Antennas
- Media Conversion - Optical and Single Pair (SPE)

Key Features

- Integrated PHY Capabilities
  - 10/100/1000BASE-T with LinkMD® cable diagnostics
  - Auto MDI/MDI-X Support
  - Support for Fast Link Up
  - Support for IEEE 802.3az Energy Efficient Ethernet (EEE)
  - Support for IEEE 802.3x Full/Half Flow & Collision Control
  - Support for IEEE 802.1Qbb Priority Flow Control
- Switch Capabilities
  - Full wire-speed, non-blocking Gigabit switch core
  - IEEE 802.1Q C-tag and S-tag VLAN support
  - 4K VLANs
  - Hashed MAC table with aging
  - Non-hashed MAC stream table
  - Source-port check
  - Layer 2 multicast
  - 8 priorities per port
  - AVB and TSN hardware support
    - IEEE 802.1Qbv Time Aware Shaping (TAS)
    - IEEE 802.1Qbu/802.3br Cut through and Preemption
    - IEEE 802.1Qav AVB traffic shaping
    - IEEE 802.1Qci Per Stream Filtering and Policing
    - IEEE 802.1Qch Cyclic Queuing and Forwarding
  - Advanced TCAM classification and VLAN push/ pop for tunneling over TSN
  - Internet Group Management (IGMPv2, IGMPv3)
  - Multicast Listener Discovery (MLDv1, MLDv2)
- Interfaces
  - 5x 10/100/1000BASE-T
  - 2x RMII/RGMII
  - 1x SGMII/SGMII+/Quad-SGMII
  - 1x SGMII/SGMII+
  - SPI, I^2C, and MIIM management interfaces
  - I^2C for SFP support and external EEPROM
- Embedded Initialization System
  - Support for configuration via Pin strapping
  - Optional I^2C EEPROM supporting internal register and switch initializations
  - Support for internal and external user-defined PHY configurations
- Network Redundancy with Hardware Assist
  - IEEE 802.1CB Frame Replication and Elimination for Reliability (FRER) with Unmanaged mode support
  - IEEE 802.1w/s Rapid and Multiple Spanning Tree (RSTP/MSTP)
  - IEC 62439-2 2016 Media Redundancy Protocol (MRP)
  - IEC 62439-3 Parallel Redundancy Protocol (PRP) Doubly Attached Node (DANP)
  - IEC 62439-3 High-Availability Seamless Redundancy (HSR) Doubly Attached Node (DANH)

*select configurations

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include -literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

TABLE OF CONTENTS

1.0 PREFACE

1.1 General Terms

TABLE 1-1: GENERAL TERMS

Term	Description
1000BASE-T	1 Gbps Ethernet over twisted pair, IEEE 802.3 compliant
100BASE-TX	100 Mbps Ethernet over twisted pair, IEEE 802.3 compliant
10BASE-T	10 Mbps Ethernet over twisted pair, IEEE 802.3 compliant
10BASE-Te	Energy-efficient version of 10BASE-T, IEEE 802.3 compliant
ACE/ACL	Access Control Entry/List
AFI	Automatic Frame Injector
ANEG	Auto-Negotiation
ATS	Asynchronous Traffic Shaping
AVB	Audio Video Bridging
BC	Basic Classifier
BIST, MEMBIST	Built In Self Test, Memory BIST
CAN	Controller Area Network
CC(M)	Continuity Check (OAM) Message
CDM	Charged Device Model
CQF	Cyclic Queuing and Forwarding
CSR	Control and Status Register
CuPHY	Copper Physical Layer
DEVCPU	CPU Port Module
DLB	Dual Leaky Bucket policing/shaping algorithm
DLR	Device Level Ring
DMA	Direct Memory Access
DP	Drop Precedence
DSA	Distributed Switch Architecture
DSCP	Differentiated Services Code Point
DWRR	Deficit Weighted Round Robin scheduling algorithm
ECC	Error Correction Code
EEE	Energy Efficient Ethernet

TABLE 1-1: GENERAL TERMS

EEPROM	Electrically Erasable Programmable Read Only Memory
ELPS	Ethernet Linear Protection Switching
ERPS	Ethernet Ring Protection Switching
ESD	Electrostatic Discharge
ESDX	Egress Service Index
FC	Flow Control
FCS	Frame Check Sequence
FDMA	Frame DMA
FDX	Full Duplex
FRER	Frame Replication and Elimination for Reliability
FW	Firmware
GMII	Gigabit Media Independent Interface
GPIO	General Purpose I/O
gPTP	Generalized PTP
HDX	Half Duplex
HBM	Human Body Model
HSR	High-availability Seamless Redundancy protocol
HW	Hardware. Refers to function implemented by digital logic.
IGMP	Internet Group Management Protocol
IRQ	Interrupt Request
ISDX	Ingress Service Index
I2C or I^2C	Inter Integrated Circuit serial management protocol, same as TWI
JTAG	Joint Test Action Group, chip/board test technology
Level- Triggered Sticky Bit	This type of status bit is set whenever the condition that it represents is asserted. The bit remains set until the condition is no longer true, and the status bit is cleared by writing a zero.
LM-StaX	Lightly Managed Microchip switch software
LRE	Link Redundancy Entity
LSDU	Link Service Data Unit
LVDS	Low Voltage Differential Signaling
MAC	Media Access Controller
MCU	Micro Controller Unit
MDI	Medium Dependent Interface

TABLE 1-1: GENERAL TERMS

MDIO / MIIM / SMI	Management Data Input/Output, MII Management,Serial Management Interface
MEF	Metro Ethernet Forum
MESA	Microchip Ethernet Switch API
MII	Media Independent Interface
MLD	Multicast Listener Device protocol (for IPv6 multicast)
MM	MAC Merge sublayer
MRP	Media Redundancy Protocol
N/A	Not Applicable
NPI	Node Processor Interface
OAM	Operations, Administration, Maintenance
PCIe	Peripheral Component Interconnect (PCI) Express
PCP	Priority Code Point
PCS	Physical Coding Sublayer
PLL	Phase Locked Loop
POD	Package Outline Drawing
(1)PPS	(one) Pulse per Second
PRBS	Pseudo Random Bit Sequence
PRP	Parallel Redundancy Protocol
PSFP	Per-Stream Filtering and Policing
PTP	Precision Time Protocol
PVT	Process, Voltage, and Temperature
QFP	Quad Flat Pack package type
QFN	Quad Flat No-lead package type
QoS	Quality of Service
RCT	Redundancy Control Trailer
RESERVED	Refers to a reserved bit field or address. Unless otherwise noted, reserved bits must always be zero for write operations. Unless otherwise noted, values are not guaranteed when reading reserved bits. Unless otherwise noted, do not read or write to reserved addresses.
RGMII	Reduced Gigabit Media Independent Interface
RMII	Reduced Media Independent Interface
RR	Round Robin scheduling algorithm
RTE	Real Time Engine

TABLE 1-1: GENERAL TERMS

SFP	1Gbps Small Form Factor pluggable transceiver
SI	Serial Interface
SLB	Single Leaky Bucket policing/shaping algorithm
SoC	System on Chip
SPI, QSPI, OSPI	Serial Peripheral Interface (Quad SPI, Octal SPI)
SQI	Signal Quality Index
SQS	Shared Queue System
TAS	Time-Aware Shaping
TCAM	Ternary Content Addressable Memory
TDR	Time Domain Reflectometer
TS	Timestamp
TSN	Time Sensitive Networking series of standards
TWI	Two-Wire Interface, same as I2C or I^2C
UART	Universal Asynchronous Receiver Transmitter
VCAP	Versatile Content Aware Processor
VRAP	Versatile Register Access Protocol
WDT	Watchdog Timer

1.2 Buffer Types

TABLE 1-2: BUFFER TYPE DESCRIPTIONS

Buffer	Description
5G_I	5G differential SerDes input
5G_O	5G differential SerDes output
AI	Analog input
AO	Analog output
AIO	Analog bidirectional
GND	Ground pin
HS_I	Variable voltage input with High Speed (RGMII compatible)
HS_O	Variable voltage output with High Speed (RGMII compatible)
IS	Schmitt-triggered input
O	Output with variable drive (10/8/4/2 mA) sink and source
ICLK	Crystal oscillator input pin
OCLK	Crystal oscillator output pin
VIS	Variable voltage Schmitt-triggered input
VO	Variable voltage output with variable drive (10/8/4/2 mA) sink and source
PU	70 kΩ (typical) internal pull-up. Unless otherwise noted in the pin description, internal pull-ups are always enabled. **Note:** Internal pull-up resistors prevent unconnected inputs from floating. Do not rely on internal resistors to drive signals external to the device. When connected to a load that must be pulled high, an external resistor must be added.
PD	70 kΩ (typical) internal pull-down. Unless otherwise noted in the pin description, internal pull-downs are always enabled. **Note:** Internal pull-down resistors prevent unconnected inputs from floating. Do not rely on internal resistors to drive signals external to the device. When connected to a load that must be pulled low, an external resistor must be added.
P	Power pin

Note: Digital signals are not 5V tolerant unless specified.

1.3 Reference Documents

1. *IEEE Standard for Ethernet*, IEEE 802.3-2022, https://ieeexplore.ieee.org/document/9844436
2. IEEE Std 1588-2008 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, https://standards.ieee.org/ieee/1588/4355/
3. IEEE Std 1588 2019 Standard for Local and Metropolitan Area Networks, Timing and Synchronization for Time-Sensitive Applications, https://standards.ieee.org/ieee/1588/6825/
4. IEEE 802.1AS-2020 Standard for Ethernet Amendment 5: Specification and Management Parameters for Interspersing Express Traffic, https://standards.ieee.org/ieee/802.1AS/7121/
5. IEEE 802.3az Energy Efficient Ethernet (EEE), https://standards.ieee.org/ieee/802.3az/4270/
6. IEEE 802.3x Full/Half Flow & Collision Control, https://standards.ieee.org/ieee/802.3x/1082/

7. IEEE 802.1Qbb Priority Flow Control, https://standards.ieee.org/ieee/802.1Qbb/4361/
8. IEEE 802.1Qbv Time Aware Shaping (TAS), https://standards.ieee.org/ieee/802.1Qbv/6068/
9. IEEE 802.1Qbu/802.3br Cut through and Preemption, https://standards.ieee.org/ieee/802.1Qbu/5464/ and https://standards.ieee.org/ieee/802.3br/5814/
10. IEEE 802.1Qav AVB traffic shaping, https://standards.ieee.org/ieee/802.1Qav/4401/
11. IEEE 802.1Qci-2017 Per-Stream Filtering and Policing, https://standards.ieee.org/ieee/802.1Qci/6159/
12. IEEE 802.1cr-2020 Asynchronous Traffic Shaping, https://standards.ieee.org/ieee/802.1Qcr/7420/
13. IEEE 802.1Qch Cyclic Queuing and Forwarding, https://standards.ieee.org/ieee/802.1Qch/6072/
14. IEEE 802.1CB Frame Replication and Elimination for Reliability (FRER), https://standards.ieee.org/ieee/802.1CB/5703/
15. IEEE 802.1w Rapid Spanning Tree (RSTP), https://standards.ieee.org/ieee/802.1w/1046/
16. IEEE 802.1s Multiple Spanning Tree (MSTP), https://standards.ieee.org/ieee/802.1s/1042/
17. IEC 62439-2 2016 Media Redundancy Protocol (MRP), https://webstore.iec.ch/en/publication/24434
18. IEC 62439-3 Parallel Redundancy Protocol (PRP) Doubly Attached Node (DANP), https://webstore.iec.ch/en/publication/64423
19. IEC 62439-3 High-Availability Seamless Redundancy (HSR) Doubly Attached Node (DANH), https://webstore.iec.ch/en/publication/64423
20. RMII Specification Revision 1.2, http://ebook.pldworld.com/_eBook/-Telecommunications,Networks-/TCPIP/RMII/rmii_rev12.pdf
21. Reduced Gigabit Media Independent Interface (RGMII) Specification Version 2.0, https://web.archive.org/web/20160303171328/http://www.hp.com/rnd/pdfs/RGMIIv2_0_final_hp.pdf
22. *Cisco QSGMII Specification*, EDCS-540123 Rev. 1.3
23. *Cisco Q-USGMII Specification*, EDCS-1155168 Rev. 4.2

2.0 INTRODUCTION

2.1 General Description

Microchip's LAN9645xF is a family of compact and cost-effective nine, seven, and five port Gigabit AVB/TSN Ethernet Switches with integrated 10/100/1000BASE-T PHYs. This family of devices can be operated in a stand-alone unmanaged system configuration or managed mode, with full Linux DSA support on a connected host.

The LAN9645xF offers up to nine switch core ports that can be allocated by the application across either four or five 10/100/1000 Mbps integrated Ethernet PHY interfaces, two SGMII/SGMII+ interfaces, two RMII/RGMII interfaces, and/or one Quad-SGMII interface. The LAN9645xF is available in nine, seven, and five switch core port configurations.

The switch core supports full line speed on each port. IEEE 802.3ad Link Aggregation is also supported, allowing multiple ports to be grouped for enhanced up-link bandwidth when shared across multiple down stream ports.

Each integrated PHY port is configurable with two LED drive outputs for direct interconnect with common Ethernet magnetic assemblies. Integrated PHYs also support LinkMD® Time Domain Reflectometry (TDR) Cable Diagnostics and Auto MDI/MDI-X, to detect and address common cabling issues.

RGMII ports offer individually configurable interfaces that support 10/100/1000 Mbps MAC or PHY mode operation for connection to user-defined Ethernet PHY transceivers or optional host processor/controller nodes. RGMII ports can also be configured for RMII mode, providing a reduced signal count at 10/100 Mbps data rates.

SGMII/SGMII+ ports offer a lower pin-count interface, supporting a variety of devices that include user-defined PHYs, SFP modules, host processors, and daisy-chained switch operation. SGMII+ ports support 2.5 Gbps data rates for higher bandwidth as well as 1 Gbps and 100FX (100 Mbps) protocols found in industrial applications. SGMII+ ports can be configured for SGMII (PHY/MAC) or 1000BASE-X SerDes operation to support a broad array of SFP modules and host equipment in managed (or unmanaged) switch configurations.

A Quad-SGMII interface supports up to four (4 x 1 Gbps) switch core ports over a single serial interface. When paired with other components from the Microchip portfolio, such as LAN8804/LAN8814, up to four additional 10/100/1000 Mbps copper PHY interfaces can be realized on a single 9-port switch.

The LAN9645xF unmanaged mode of operation supports Layer-2 Ethernet switch features such as auto-negotiation, SerDes configuration, MAC address learning and aging, flow control, Energy-Efficient Ethernet, loop prevention, and E2E transparent clock.

When paired with external EEPROM devices, the LAN9645xF allows initialization of internal registers and external PHY devices during boot.

The LAN9645xF offers port management support with I^2C, SPI, and MIIM peripheral interfaces. The user can initialize these peripherals according to the type of port device and mode selected. Host modes enable initialization of external user-selected PHYs or interrogation of SFP capabilities. Alternatively, device modes support switch management and monitoring by an external host.

In addition to I^2C, SPI, and MIIM management ports, In-Band Management (IBA) can be used for switch configuration on any Ethernet switch port using the Versatile Register Access Protocol (VRAP).

The LAN9645xF device is generally designed for edge networks such as managed DIN rail, wall-mounted, and IP67-type switches, and is tailored for a variety of applications where parts of the traffic require deterministic latency.

LAN9645xF supports IEEE 1588-2019 (PTP) time synchronization with time stamping in multiple domains in support of Working Clock and Global Time distribution.

The LAN9645xF is ideal for industrial and other high-reliability applications, supporting IEEE 802.1w Rapid Spanning Tree (RSTP), IEEE 802.1CB Frame Replication, Elimination for Redundancy (FRER), and IEC 62439-2 Media Redundancy Protocols (MRP). FRER is supported in managed or unmanaged configurations. LAN9645xF also supports IEC 62439-3 Parallel Redundancy (PRP) and IEC 62439-3 High-Availability Seamless Redundancy (HSR) protocols in Doubly Attached Node (DANP/DANH) mode.

LAN9645xF is also compatible with TSN domain protection though IEEE 802.1Qci Per Stream Filtering and Policing, advanced classification rules, and tunneling by adding VLAN tags. Each egress port supports eight priorities. Prior to scheduling, traffic can be shaped using IEEE 802.1Qav Credit Based Shaping or IEEE 802.1Qbv Time Aware Shaping. Low latency is possible though architecture, cut-through, and IEEE 802.1Qbu/802.3br Preemption.

LAN9645xF supports an industrial (-40°C to +85°C) temperature range (for select configurations) and is available in 128-pin TQFP (14 x 14 mm) or 156-pin VQFN-DR (12 x 12 mm) packages.

TABLE 2-1: LAN96459/7/5F FEATURE MATRIX

Part Number	Port Count	Allowable Number of CuPHY	QSGMII #1 (Note 1)	SGMII #1 (Note 2)	SGMII #2 (Note 3)	RGMII #1	RGMII #2	TSN	Description
Max	**9**	**5**	**x4**	**1**	**1**	**1**	**1**	**-**	**-**
LAN96459F	9	4-5	x	x	x	x	x	x	9-Port, with TSN on all interfaces
LAN96457F	7	2-5	x	x	x	x	x	x	7-Port, with TSN on all selectable interfaces
LAN96455F	5	0-5	x	x	x	x	x	x	5-Port, with TSN on all selectable interfaces

Note 1: QSGMII is bonded to the same physical SGMII #1 pins and conveys up to 4x 1G switch ports on a single high-speed interface.

2: QSGMII mode is mutually exclusive to SGMII #1 or SGMII #2 operation.

3: Each SGMII interface supports 1G or 2.5G(+) mode individually. Switch core supports full line speed throughput at either speed.

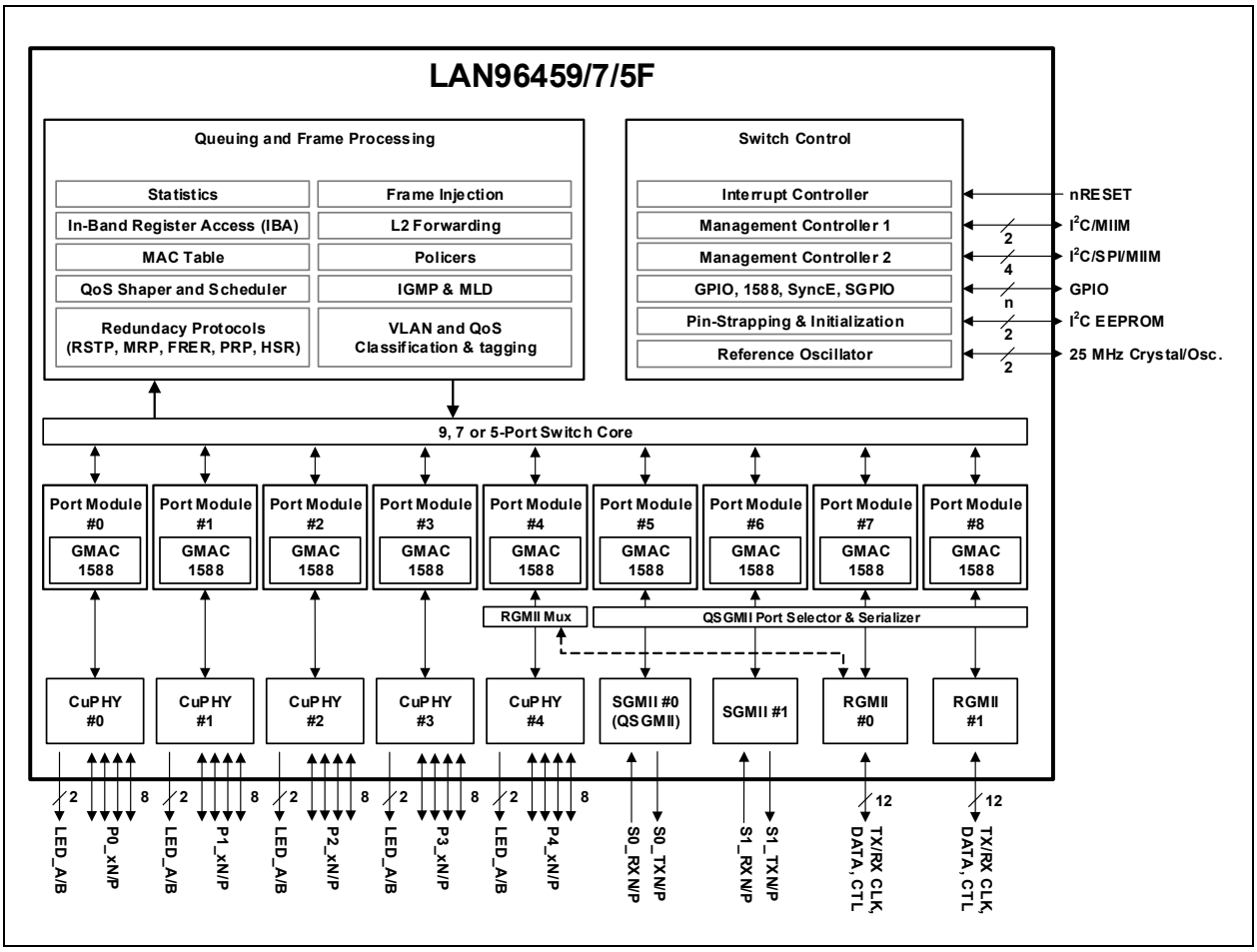
Table 2-2 provides examples of major port configurations based on the part number's port count.

TABLE 2-2: LAN96459/7/5F MAJOR PORT CONFIGURATIONS

Part Number	Port count	Major Port Configurations
LAN96459F	9	5x CuPHY + 2x 2.5G SGMII + 2x RGMII
		5x CuPHY + 1x QSGMII
		4x CuPHY + 1x QSGMII + 1x RGMII
LAN96457F	7	5x CuPHY + 2x 2.5G SGMII
		5x CuPHY + 2x RGMII
		5x CuPHY + 1x 2.5G SGMII + 1x RGMII
		4x CuPHY + 1x 2.5G SGMII + 2x RGMII
		4x CuPHY + 2x 2.5G SGMII + 1x RGMII
		3x CuPHY + 2x 2.5G SGMII + 2x RGMII
		3x CuPHY + 1x QSGMII
		2x CuPHY + 1x QSGMII + 1x RGMII
LAN96455F	5	5x CuPHY
		4x CuPHY + 1x 2.5G SGMII
		4x CuPHY + 1x RGMII
		3x CuPHY + 2x 2.5G SGMII
		3x CuPHY + 2x RGMII
		3x CuPHY + 1x 2.5G SGMII + 1x RGMII
		2x CuPHY + 1x 2.5G SGMII + 2x RGMII
		2x CuPHY + 2x 2.5G SGMII + 1x RGMII
		1x CuPHY + 2x 2.5G SGMII + 2x RGMII

An internal block diagram of the LAN96459F (nine port configuration) is shown in Figure 2-1.

FIGURE 2-1: LAN96459F SYSTEM BLOCK DIAGRAM



2.2 Typical Applications

FIGURE 2-2: LAN96459F (WITH LAN8804/LAN8814) - UNMANAGED 9-PORT 10/100/1000BASE-T SWITCH SYSTEM

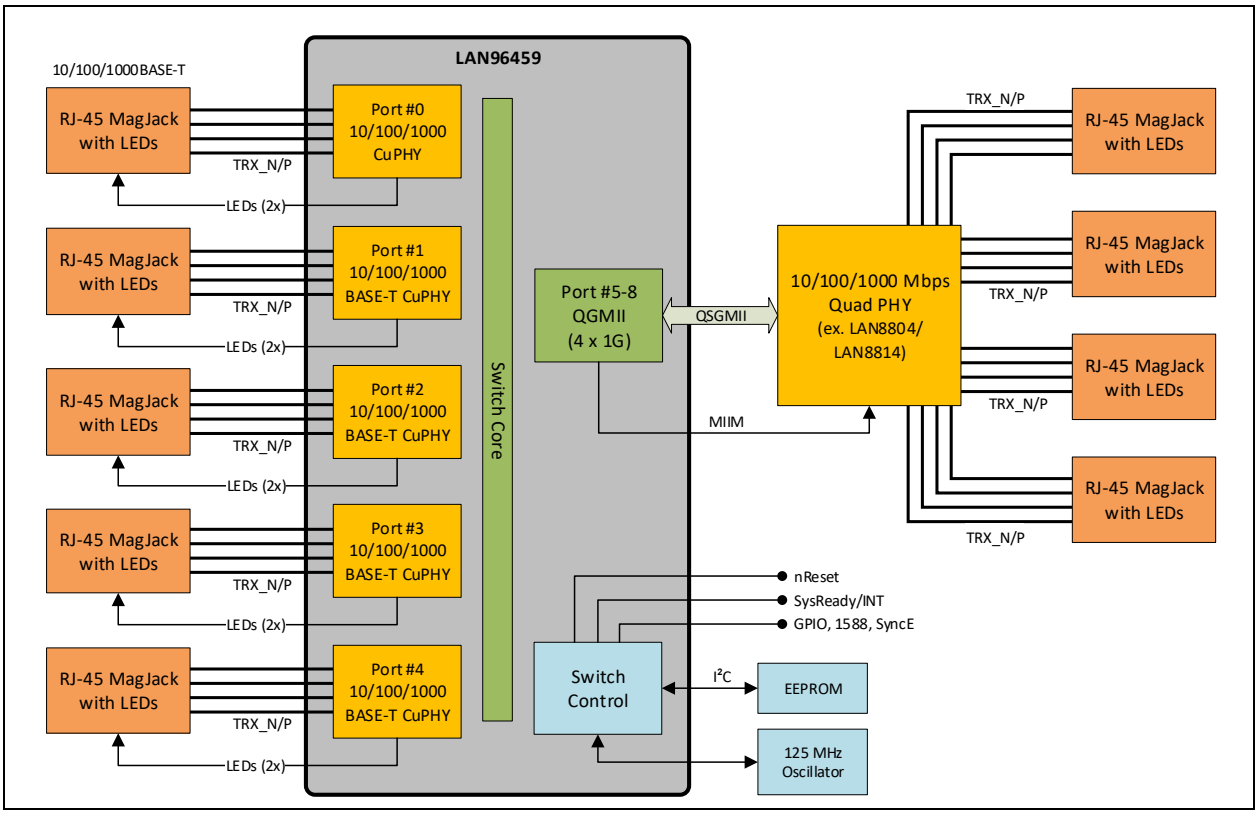


FIGURE 2-3: LAN96459F (WITH LAN8870/LAN8872 &VSC8531-02) TSN MANAGED 9-PORT TSN 10/100/1000BASE-T + T1 SWITCH SYSTEM

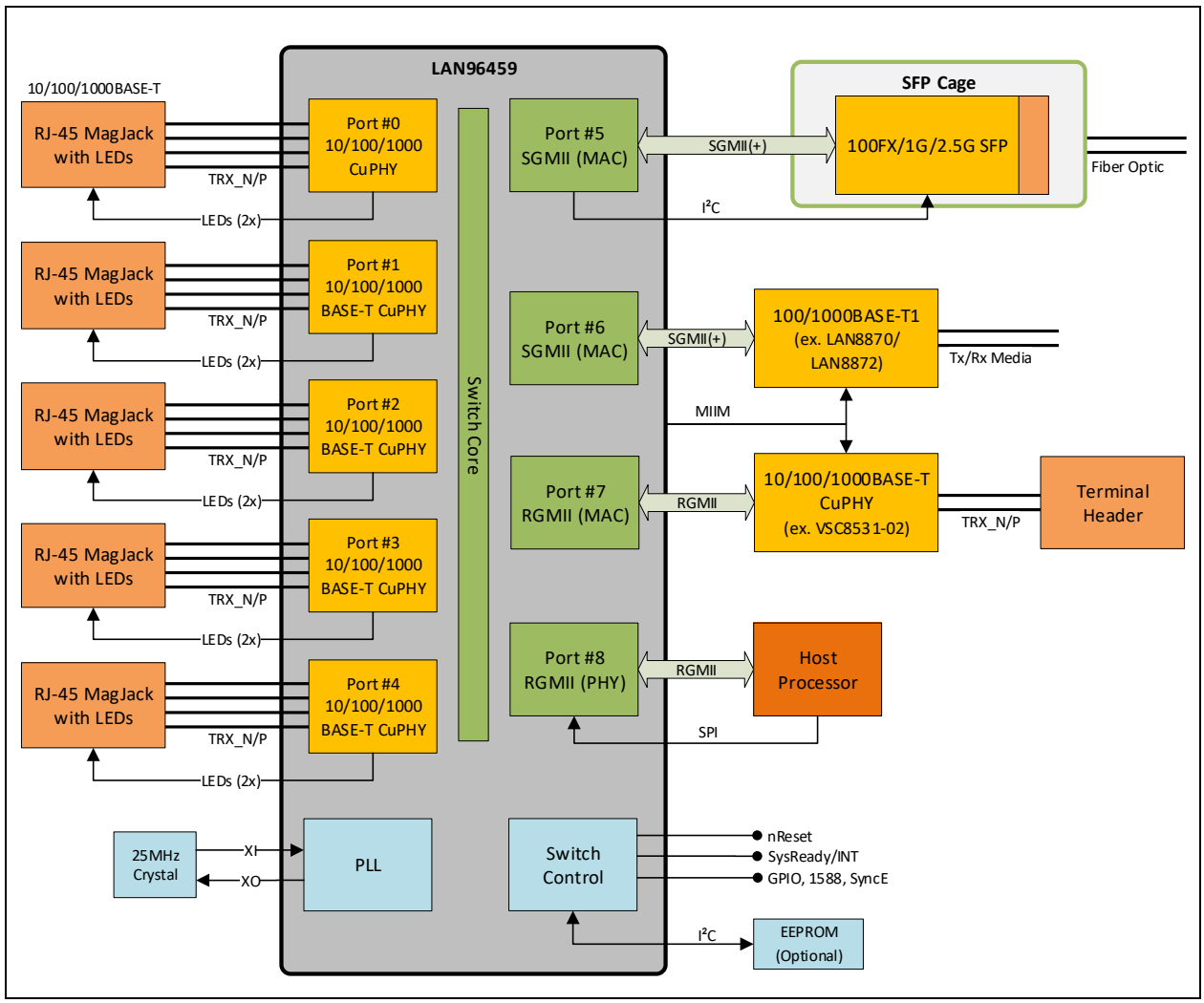


FIGURE 2-4: LAN96459F (WITH LAN8870/LAN8872 & LAN8671) - UNMANAGED 9-PORT 10/100/1000BASE-T + T1 + T1S + DUAL SPF SWITCH SYSTEM

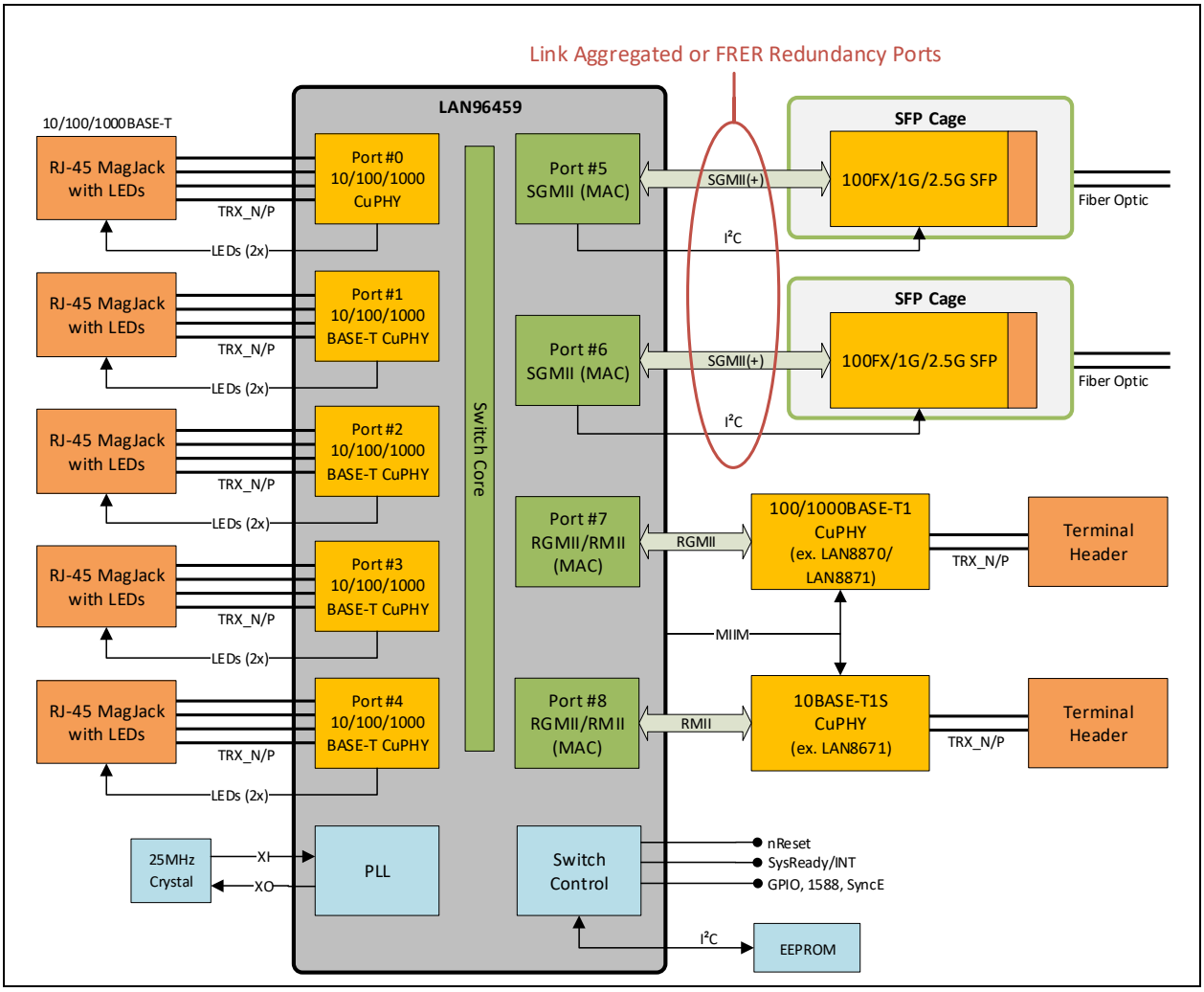
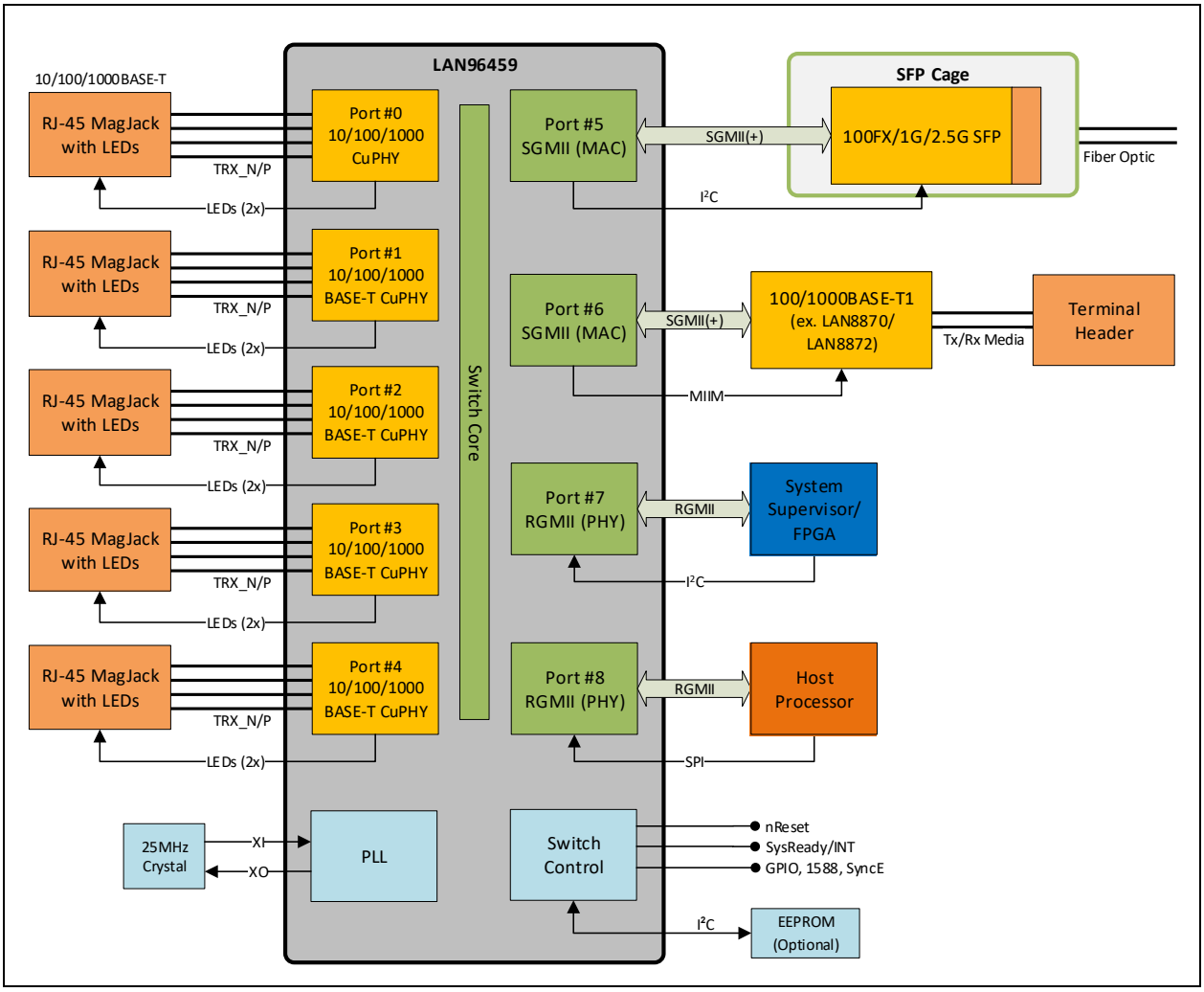


FIGURE 2-5: LAN96459F (WITH LAN8870/LAN8872 & LAN8671) - TSN MANAGED 9-PORT 10/100/1000BASE-T + T1 + DUAL NODE WITH SPF UPLINK SWITCH SYSTEM



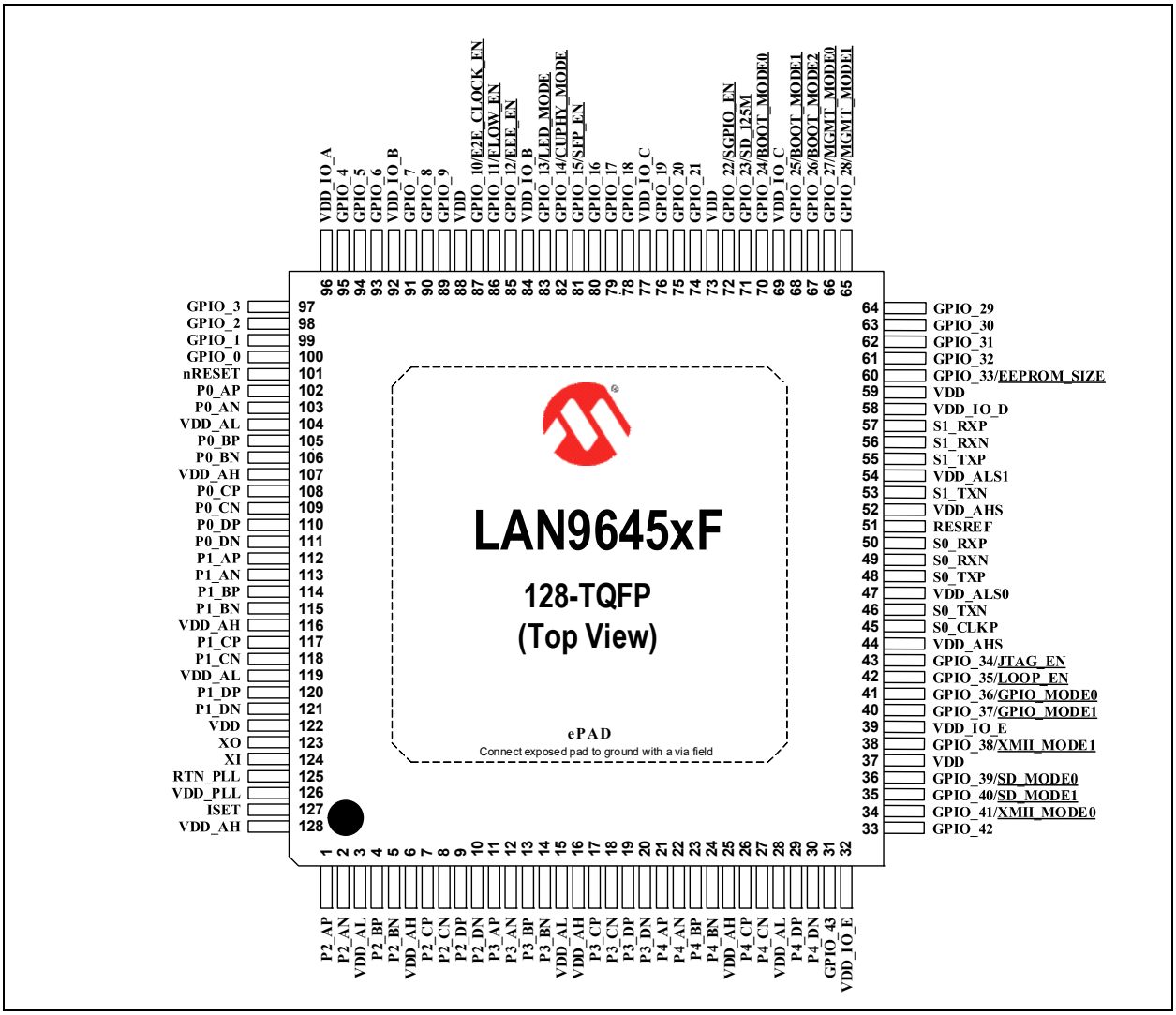
3.0 PIN DESCRIPTIONS AND CONFIGURATION

This chapter includes the following sections:

3.1 Pin Assignments

3.1.1 128-TQFP PIN ASSIGNMENTS

FIGURE 3-1: 128-TQFP PIN ASSIGNMENTS (TOP VIEW)



Note:	Configuration straps are identified by an underlined symbol name. Strap signals must have an external pull-up or pull-down resistor, regardless of whether they are connected to a load or not.

TABLE 3-1: 128-PIN TQFP ASSIGNMENTS

Pin Num.	Pin Name	Pin Num.	Pin Name
1	P2_AP	65	GPIO_28/MGMT_MODE1
2	P2_AN	66	GPIO_27/MGMT_MODE0
3	VDD_AL	67	GPIO_26/BOOT_MODE2
4	P2_BP	68	GPIO_25/BOOT_MODE1
5	P2_BN	69	VDD_IO_C
6	VDD_AH	70	GPIO_24/BOOT_MODE0
7	P2_CP	71	GPIO_23/SD_125M
8	P2_CN	72	GPIO_22/SGPIO_EN
9	P2_DP	73	VDD
10	P2_DN	74	GPIO_21
11	P3_AP	75	GPIO_20
12	P3_AN	76	GPIO_19
13	P3_BP	77	VDD_IO_C
14	P3_BN	78	GPIO_18
15	VDD_AL	79	GPIO_17
16	VDD_AH	80	GPIO_16
17	P3_CP	81	GPIO_15/SFP_EN
18	P3_CN	82	GPIO_14/CUPHY_MODE
19	P3_DP	83	GPIO_13/LED_MODE
20	P3_DN	84	VDD_IO_B
21	P4_AP	85	GPIO_12/EEE_EN
22	P4_AN	86	GPIO_11/FLOW_EN
23	P4_BP	87	GPIO_10/E2E_CLOCK_EN
24	P4_BN	88	VDD
25	VDD_AH	89	GPIO_9
26	P4_CP	90	GPIO_8
27	P4_CN	91	GPIO_7
28	VDD_AL	92	VDD_IO_B
29	P4_DP	93	GPIO_6
30	P4_DN	94	GPIO_5
31	GPIO_43	95	GPIO_4
32	VDD_IO_E	96	VDD_IO_A
33	GPIO_42	97	GPIO_3
34	GPIO_41/XMII_MODE0	98	GPIO_2
35	GPIO_40/SD_MODE1	99	GPIO_1
36	GPIO_39/SD_MODE0	100	GPIO_0
37	VDD	101	nRESET
38	GPIO_38/XMII_MODE1	102	P0_AP
39	VDD_IO_E	103	P0_AN
40	GPIO_37/GPIO_MODE1	104	VDD_AL
41	GPIO_36/GPIO_MODE0	105	P0_BP
42	GPIO_35/LOOP_EN	106	P0_BN
43	GPIO_34/JTAG_EN	107	VDD_AH
44	VDD_AHS	108	P0_CP

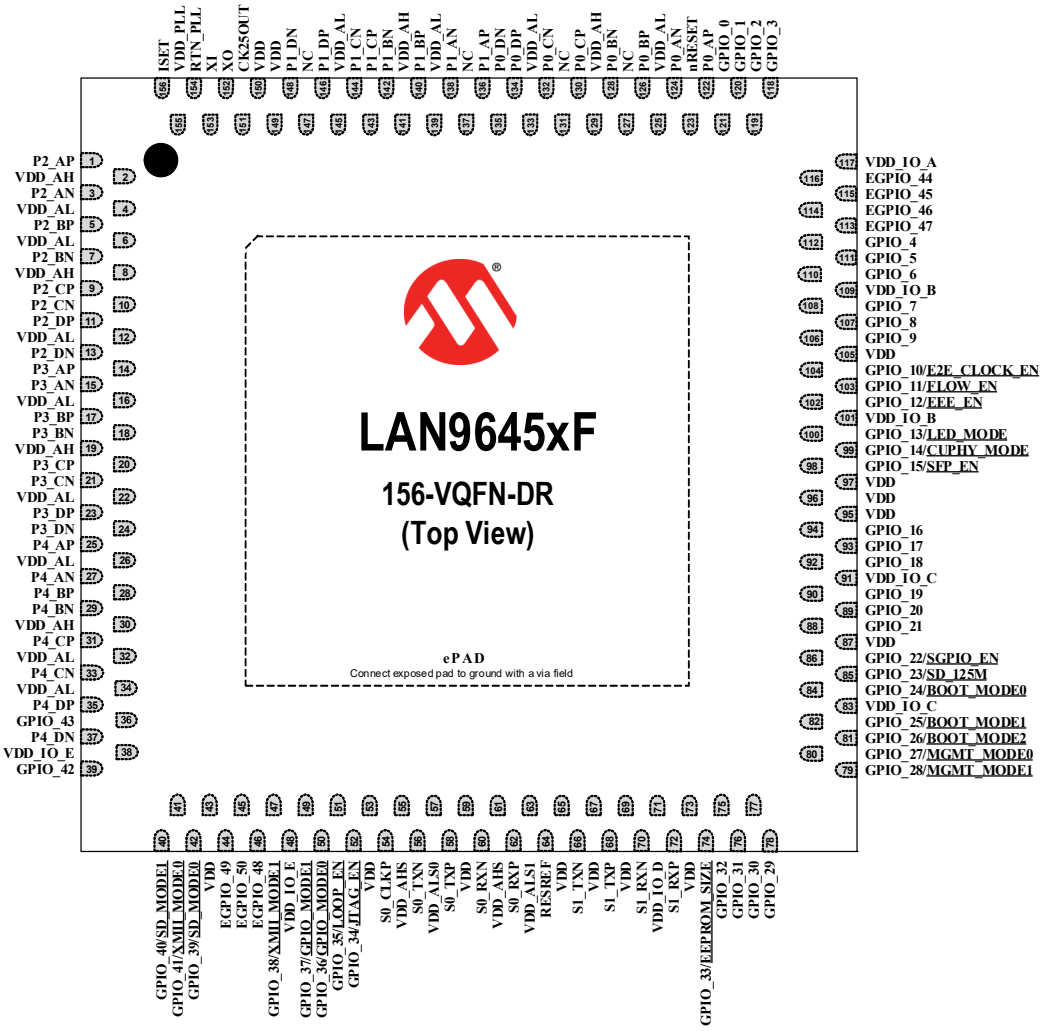
TABLE 3-1: 128-PIN TQFP ASSIGNMENTS (CONTINUED)

Pin Num.	Pin Name	Pin Num.	Pin Name
45	S0_CLKP	109	P0_CN
46	S0_TXN	110	P0_DP
47	VDD_ALS0	111	P0_DN
48	S0_TXP	112	P1_AP
49	S0_RXN	113	P1_AN
50	S0_RXP	114	P1_BP
51	RESREF	115	P1_BN
52	VDD_AHS	116	VDD_AH
53	S1_TXN	117	P1_CP
54	VDD_ALS1	118	P1_CN
55	S1_TXP	119	VDD_AL
56	S1_RXN	120	P1_DP
57	S1_RXP	121	P1_DN
58	VDD_IO_D	122	VDD
59	VDD	123	XO
60	GPIO_33/EEPROM_SIZE	124	XI
61	GPIO_32	125	RTN_PLL
62	GPIO_31	126	VDD_PLL
63	GPIO_30	127	ISET
64	GPIO_29	128	VDD_AH
Exposed Pad (**ePAD**) must be connected to ground.			

Note: Configuration straps are identified by an underlined symbol name. Strap signals must have an external pull-up or pull-down resistor, regardless of whether they are connected to a load or not.

3.1.2 156-PIN VQFN-DR ASSIGNMENTS

FIGURE 3-2: 156-PIN VQFN-DR ASSIGNMENTS (TOP VIEW)



Note: Configuration straps are identified by an underlined symbol name. Strap signals must have an external pull-up or pull-down resistor, regardless of whether they are connected to a load or not.

TABLE 3-2: 156-PIN VQFN-DR ASSIGNMENTS

Pin Num.	Pin Name	Pin Num.	Pin Name
1	P2_AP	79	GPIO_28/MGMT_MODE1
2	VDD_AH	80	GPIO_27/MGMT_MODE0
3	P2_AN	81	GPIO_26/BOOT_MODE2
4	VDD_AL	82	GPIO_25/BOOT_MODE1
5	P2_BP	83	VDD_IO_C
6	VDD_AL	84	GPIO_24/BOOT_MODE0
7	P2_BN	85	GPIO_23/SD_125M
8	VDD_AH	86	GPIO_22/SGPIO_EN
9	P2_CP	87	VDD
10	P2_CN	88	GPIO_21
11	P2_DP	89	GPIO_20
12	VDD_AL	90	GPIO_19
13	P2_DN	91	VDD_IO_C
14	P3_AP	92	GPIO_18
15	P3_AN	93	GPIO_17
16	VDD_AL	94	GPIO_16
17	P3_BP	95	VDD
18	P3_BN	96	VDD
19	VDD_AH	97	VDD
20	P3_CP	98	GPIO_15/SFP_EN
21	P3_CN	99	GPIO_14/CUPHY_MODE
22	VDD_AL	100	GPIO_13/LED_MODE
23	P3_DP	101	VDD_IO_B
24	P3_DN	102	GPIO_12/EEE_EN
25	P4_AP	103	GPIO_11/FLOW_EN
26	VDD_AL	104	GPIO_10/E2E_CLOCK_EN
27	P4_AN	105	VDD
28	P4_BP	106	GPIO_9
29	P4_BN	107	GPIO_8
30	VDD_AH	108	GPIO_7
31	P4_CP	109	VDD_IO_B
32	VDD_AL	110	GPIO_6
33	P4_CN	111	GPIO_5
34	VDD_AL	112	GPIO_4
35	P4_DP	113	EGPIO_47
36	GPIO_43	114	EGPIO_46
37	P4_DN	115	EGPIO_45
38	VDD_IO_E	116	EGPIO_44
39	GPIO_42	117	VDD_IO_A
40	GPIO_40/SD_MODE1	118	GPIO_3
41	GPIO_41/XMII_MODE0	119	GPIO_2
42	GPIO_39/SD_MODE0	120	GPIO_1
43	VDD	121	GPIO_0
44	EGPIO_49	122	P0_AP

TABLE 3-2: 156-PIN VQFN-DR ASSIGNMENTS (CONTINUED)

Pin Num.	Pin Name	Pin Num.	Pin Name
45	EGPIO_50	123	nRESET
46	EGPIO_48	124	P0_AN
47	GPIO_38/XMII_MODE1	125	VDD_AL
48	VDD_IO_E	126	P0_BP
49	GPIO_37/GPIO_MODE1	127	NC
50	GPIO_36/GPIO_MODE0	128	P0_BN
51	GPIO_35/LOOP_EN	129	VDD_AH
52	GPIO_34/JTAG_EN	130	P0_CP
53	VDD	131	NC
54	S0_CLKP	132	P0_CN
55	VDD_AHS	133	VDD_AL
56	S0_TXN	134	P0_DP
57	VDD_ALS0	135	P0_DN
58	S0_TXP	136	P1_AP
59	VDD	137	NC
60	S0_RXN	138	P1_AN
61	VDD_AHS	139	VDD_AL
62	S0_RXP	140	P1_BP
63	VDD_ALS1	141	VDD_AH
64	RESREF	142	P1_BN
65	VDD	143	P1_CP
66	S1_TXN	144	P1_CN
67	VDD	145	VDD_AL
68	S1_TXP	146	P1_DP
69	VDD	147	NC
70	S1_RXN	148	P1_DN
71	VDD_IO_D	149	VDD
72	S1_RXP	150	VDD
73	VDD	151	CK25OUT
74	GPIO_33/EEPROM_SIZE	152	XO
75	GPIO_32	153	XI
76	GPIO_31	154	RTN_PLL
77	GPIO_30	155	VDD_PLL
78	GPIO_29	156	ISET
Exposed Pad (**ePAD**) must be connected to ground.			

Note: Configuration straps are identified by an underlined symbol name. Strap signals must have an external pull-up or pull-down resistor, regardless of whether they are connected to a load or not.

3.2 Pin Descriptions

This section contains descriptions of the various LAN9645xF pins. GPIO alternate functions are detailed in **Section 3.3 "GPIO Alternate Functions"**. Configuration straps are detailed in **Section 3.4 "Configuration Straps"**. Buffer type definitions are detailed in Section 1.2, Buffer Types.

The "**n**" symbol in the signal name indicates that the active, or asserted, state occurs when the signal is at a low voltage level. For example, **nRESET** indicates that the reset signal is active low. When "**n**" is not present at the beginning of the signal name, the signal is asserted when at the high voltage level.

TABLE 3-3: PIN ASSIGNMENTS

Name	Symbol	Buffer Type	Description
10/100/1000BASE-T Ethernet PHY Pins			
Ports 4-0 Ethernet PHY Channel A Positive	**P[4:0]_AP**	AIO	Ports 4-0 10/100/1000BASE-T PHY Channel A Positive Differential Pair
Ports 4-0 Ethernet PHY Channel A Negative	**P[4:0]_AN**	AIO	Ports 4-0 10/100/1000BASE-T PHY Channel A Negative Differential Pair
Ports 4-0 Ethernet PHY Channel B Positive	**P[4:0]_BP**	AIO	Ports 4-0 10/100/1000BASE-T PHY Channel B Positive Differential Pair
Ports 4-0 Ethernet PHY Channel B Negative	**P[4:0]_BN**	AIO	Ports 4-0 10/100/1000BASE-T PHY Channel B Negative Differential Pair
Ports 4-0 Ethernet PHY Channel C Positive	**P[4:0]_CP**	AIO	Ports 4-0 10/100/1000BASE-T PHY Channel C Positive Differential Pair
Ports 4-0 Ethernet PHY Channel C Negative	**P[4:0]_CN**	AIO	Ports 4-0 10/100/1000BASE-T PHY Channel C Negative Differential Pair
Ports 4-0 Ethernet PHY Channel D Positive	**P[4:0]_DP**	AIO	Ports 4-0 10/100/1000BASE-T PHY Channel D Positive Differential Pair
Ports 4-0 Ethernet PHY Channel D Negative	**P[4:0]_DN**	AIO	Ports 4-0 10/100/1000BASE-T PHY Channel D Negative Differential Pair
Ethernet PHY Current Calibration	**ISET**	AI	Current calibration reference resistor connection pin for Ethernet PHYs. For proper operation, this pin must be connected to ground through a 6.0 kΩ 1% resistor.
SerDes Interface Pins			
SerDes 1-0 Data Input Positive	**S[1:0]_RXP**	AI	SerDes 1-0 Data Input Positive Differential Pair

TABLE 3-3: PIN ASSIGNMENTS

SerDes 1-0 Data Input Negative	**S[1:0]_RXN**	AI	SerDes 1-0 Data Input Negative Differential Pair
SerDes 1-0 Data Output Positive	**S[1:0]_TXP**	AO	SerDes 1-0 Data Output Positive Differential Pair
SerDes 1-0 Data Output Negative	**S[1:0]_TXN**	AO	SerDes 1-0 Data Output Negative Differential Pair
SerDes Reference Resistor	**RESREF**	AI	Reference resistor connection pin for SerDes interfaces. For proper operation, this pin must be connected to ground through a 200Ω 1% resistor
GPIOs			
GPIO 43-34, 29-28, 3-0	**GPIO_[43:34] GPIO_[29:28]] GPIO_[3:0]**	VIS/VO (PU)	General Purpose I/Os. These GPIOs support Alternate Functions. The GPIO alternate functions are described in Section 3.3, "GPIO Alternate Functions".
GPIO 33-30	**GPIO_[33:30]**	IS/O (PU)	General Purpose I/Os. These GPIOs support Alternate Functions. The GPIO alternate functions are described in Section 3.3, "GPIO Alternate Functions".
GPIO 27-4	**GPIO_[27:4]**	HS_I/ HS_O (PU)	General Purpose I/Os. These GPIOs support Alternate Functions. The GPIO alternate functions are described in Section 3.3, "GPIO Alternate Functions".
EGPIO 50-44 (156-Pin VQFN-DR Package Only)	**EGPIO_[50:44]**	VIS/VO (PU)	Extended General Purpose I/Os 0-6 (available in 156-Pin VQFN-DR package only). These GPIOs support Alternate Functions. The EGPIO alternate functions are described in Section 3.3, "GPIO Alternate Functions".
Miscellaneous Pins			
Chip Reset	**nRESET**	VIS (PU)	Active-low chip reset.
Crystal Clock / Oscillator Input	**XI**	ICLK	Crystal clock input. When using a crystal, this input is connected to one lead of the crystal. When using an oscillator, this pin is the input from the oscillator.
Crystal Clock / Oscillator Output	**XO**	OCLK	Crystal clock output. When using a crystal, this output is connected to one lead of the crystal. When using an oscillator, this pin is left unconnected.
Clock input	**S0_CLKP**	AI	125 MHz single-ended reference clock oscillator input.

TABLE 3-3: PIN ASSIGNMENTS

Clock output (156-Pin VQFN-DR Package Only)	**CK25OUT**	VO	25 MHz output clock from PLL (available in 156-Pin VQFN-DR package only).
No Connect	**NC**	-	For normal operation, this pin must be left unconnected.
Configuration Strap Pins			
Boot mode selector	**BOOT_MODE[2:0]**	HS_I/ HS_O (PU)	This configuration strap defines the booting interface for the internal controller. Refer to Section 3.4.1.1, Boot Mode Selector (BOOT_MODE[2:0]) for additional information. **Note:** **BOOT_MODE[2:0]** are used in both managed and unmanaged modes.
Management interface mode selector	**MGMT_MODE[1:0]**	Note 3-1 (PU)	This configuration strap defines which management interfaces are exposed on GPIOs. Refer to Section 3.4.1.2, Management Interface Mode Selector (MGMT_MODE[1:0]) for additional information. **Note:** **MGMT_MODE[1:0]** are used in both managed and unmanaged modes. **Note 3-1** **MGMT_MODE0** uses buffer types HS_I/ HS_O. **MGMT_MODE1** uses buffer types VIS/VO.
JTAG enable	**JTAG_EN**	VIS/VO (PU)	This configuration strap enables the JTAG interface. Refer to Section 3.4.1.3, JTAG Enable (JTAG_EN) for additional information. **Note:** **JTAG_EN** is used in both managed and unmanaged modes.
SerDes 125 MHz selector	**SD_125M**	HS_I/ HS_O (PU)	This configuration strap enables a 125 MHz reference clock input to SerDes. Refer to Section 3.4.1.4, SerDes 125 MHz Selector (SD_125M) for additional information. **Note:** **SD_125M** is used in both managed and unmanaged modes. **Note:** If **SD_125M** is strapped high, **SGPIO_EN** must be strapped low. See Section 3.4.2.10, SGPIO Selector (SGPIO_EN) for further details.
Loop prevention enable	**LOOP_EN**	VIS/VO (PU)	This configuration strap enables the loop prevention feature. Refer to Section 3.4.2.1, Loop Prevention Enable (LOOP_EN) for additional information. **Note:** **LOOP_EN** is used in unmanaged mode only.
End-to-end transparent clock enable	**E2E_CLOCK_EN**	HS_I/ HS_O (PU)	This configuration strap enables the end-to-end transparent clock feature. Refer to Section 3.4.2.2, End-to-end Transparent Clock Enable (E2E_CLOCK_EN) for additional information. **Note:** **E2E_CLOCK_EN** is used in unmanaged mode only.

TABLE 3-3: PIN ASSIGNMENTS

Flow control enable	**FLOW_EN**	HS_I/ HS_O (PU)	This configuration strap enables flow control auto-negotiation on all ports. Refer to Section 3.4.2.3, Flow Control Enable (FLOW_EN) for additional information. **Note:** **FLOW_EN** is used in unmanaged mode only.
EEE enable	**EEE_EN**	HS_I/ HS_O (PU)	This configuration strap enables Energy Efficient Ethernet on CuPHY ports. Refer to Section 3.4.2.4, Energy Efficient Ethernet Enable (EEE_EN) for additional information. **Note:** **EEE_EN** is used in unmanaged mode only.
LED mode selector	**LED_MODE**	HS_I/ HS_O (PU)	This configuration strap defines the LED operation modes. Refer to Section 3.4.2.5, LED Mode Selector (LED_MODE) for additional information. **Note:** **LED_MODE** is used in unmanaged mode only.
CuPHY mode selector	**CUPHY_MODE**	HS_I/ HS_O (PU)	This configuration strap defines if 1G speed is advertised for integrated CuPHYs. Refer to Section 3.4.2.6, CuPHY Mode Selector (CUPHY_MODE) for additional information. **Note:** **CUPHY_MODE** is used in unmanaged mode only.
2.5G SFP speed detect enable	**SFP_EN**	HS_I/ HS_O (PU)	When enabled, if a cable is plugged in but a 2.5G SFP link-up is not detected when 2.5G SFP is being used, the device will attempt to link at lower speeds. Refer to Section 3.4.2.7, 2.5G SFP Speed Detect Enable (SFP_EN) for additional information. **Note:** **SFP_EN** is used in unmanaged mode only.
EEPROM Size selector	**EEPROM_SIZE**	IS/O (PU)	This configuration strap defines the addressing size of the I^2C EEPROM. Refer to Section 3.4.2.8, EEPROM Size Selector (EEPROM_SIZE) for additional information. **Note:** **EEPROM_SIZE** is used in unmanaged mode only.
GPIO mode selector	**GPIO_MODE[1:0]**	VIS/VO (PU)	These configuration straps select one of the four possible GPIO modes (A, B, C, or D, based on available interfaces). Refer to Section 3.4.2.9, GPIO Mode Selector (GPIO_MODE[1:0]) for additional information. **Note:** **GPIO_MODE[1:0]** are used in unmanaged mode only.
SGPIO selector	**SGPIO_EN**	HS_I/ HS_O (PU)	In GPIO Mode A, this configuration strap controls the SFP signals that can be connected using either SGPIOs or regular GPIOs. For GPIO Modes B, C, or D, **SGPIO_EN** is unused. Refer to Section 3.4.2.10, SGPIO Selector (SGPIO_EN) for additional information. **Note:** **SGPIO_EN** is used in unmanaged mode only.

TABLE 3-3: PIN ASSIGNMENTS

SerDes interface mode selector	**SD_MODE[1:0]**	VIS/VO (PU)	These configuration straps define if SerDes operates in SGMII mode or in SFP mode. Refer to Section 3.4.2.11, SerDes Interface Mode SELECTOR (SD_MODE[1:0]) for additional information. **Note:** **SD_MODE[1:0]** are used in unmanaged mode only.
RGMII/RMII interface mode selector	**XMII_MODE[1:0]**	VIS/VO (PU)	These configuration straps select between RGMII and RMII interface modes. The RGMII/RMII selection options are dependent on the GPIO mode determined by the **GPIO_MODE[1:0]** strap settings. Refer to Section 3.4.2.12, RGMII/RMII Interface Mode Selector (XMII_MODE[1:0]) for additional information. **Note:** **XMII_MODE[1:0]** are used in unmanaged mode only.
Power and Ground Pins			
+1.15V SerDes 0 TX Power Supply	**VDD_ALS0**	P	+1.15V SerDes 0 TX power supply
+1.15V SerDes 1 TX Power Supply	**VDD_ALS1**	P	+1.15V SerDes 1 TX power supply
+1.15V PLL Power Supply	**VDD_PLL**	P	+1.15V PLL power supply
+2.5/3.3V Ethernet PHY and SerDes Analog High Power Supply	**VDD_AH**	P	+2.5/3.3V Ethernet PHY and SerDes analog high power supply **Note:** The voltage selected (2.5V or 3.3V) must also be used on the VDD_AHS supply.
+2.5/3.3V SerDes Analog High Power Supply	**VDD_AHS**	P	+2.5/3.3V SerDes analog high power supply **Note:** The voltage selected (2.5V or 3.3V) must also be used on the VDD_AH supply.
+1.15V Ethernet PHY Analog Low Power Supply	**VDD_AL**	P	+1.15V Ethernet PHY analog low power supply
+1.15V Core Power Supply	**VDD**	P	+1.15V Core power supply
+1.8/2.5/3.3V I/O Domain A Power Supply	**VDD_IO_A**	P	+1.8/2.5/3.3V I/O Domain A power supply
+1.8/2.5/3.3V I/O Domain B Power Supply	**VDD_IO_B**	P	+1.8/2.5/3.3V I/O Domain B power supply
+1.8/2.5/3.3V I/O Domain C Power Supply	**VDD_IO_C**	P	+1.8/2.5/3.3 I/O Domain C power supply
+3.3V I/O Domain D Power Supply	**VDD_IO_D**	P	+3.3V I/O Domain D power supply **Note:** Domain D power supply is fixed to +3.3V.
+1.8/2.5/3.3V I/O Domain E Power Supply	**VDD_IO_E**	P	+1.8/2.5/3.3V I/O Domain E power supply

TABLE 3-3: PIN ASSIGNMENTS

PLL Return	**RTN_PLL**	P	PLL return. Do not connect to ground, but directly to the PLL decoupling capacitor.
Ground	**ePAD**	GND	Ground

3.3 GPIO Alternate Functions

The LAN9645xF General Purpose I/O Pins may be configured as alternate functions. The number of GPIOs varies by package. The 128-Pin TQFP package features 44 GPIOs. The 156-Pin VQFN-DR package features 51 GPIOs, which includes 7 Extended GPIOs exclusive to this package.

In managed mode, alternate GPIO functions can be set via internal register settings. Refer to Section 3.3.1, "GPIO Alternate Functions in Managed Mode" for a list all available GPIO alternate functions in a managed switch mode.

In unmanaged mode, alternate GPIO functions can be set via the **GPIO_MODE[1:0]** configuration strapping pins. Refer to Section 3.3.2, GPIO Alternate Functions in Unmanaged Mode for a list of all available GPIO alternate functions in an unmanaged switch mode.

Detailed descriptions of the GPIO alternate functions are provided in Section 3.3.3, GPIO Alternate Function Descriptions.

3.3.1 GPIO ALTERNATE FUNCTIONS IN MANAGED MODE

In manged mode, GPIO alternate functions are set using internal register settings. Available functions are shown in Table 3-4.

Note: **EGPIO_[55:44]** are only available in the 156-Pin VQFN-DR package, respectively.

TABLE 3-4: GPIO ALTERNATE FUNCTION ASSIGNMENTS IN MANAGED MODE

GPIO	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	Manual
GPIO_0	SI_DI	SI_SLV_DI	I2C_SLV_SDA	MIIM_SLV_ADDR	UART_RXD	MIIM1_MDC	-
GPIO_1	SI_DO	SI_SLV_DO	I2C_SLV_SCL	MIIM_SLV_MDC	UART_TXD	MIIM1_MDIO	-
GPIO_2	SI_nCS	SI_SLV_nCS	I2C1_SDA	-	-	-	-
GPIO_3	SI_CLK	SI_SLV_CLK	I2C1_SCL	MIIM_SLV_MDIO	-	-	-
GPIO_4	RGMII1_RXD0	I2C0_SCL2	I2C1_SDA	-	-	SI_SLV_DI	-
GPIO_5	RGMII1_RXD1	I2C0_SCL3	I2C1_SCL	-	-	SI_SLV_DO	-
GPIO_6	RGMII1_RXD2-RXER	I2C0_SCL4	-	-	-	SI_SLV_nCS	-
GPIO_7	RGMII1_RXD3	I2C0_SCL5	SD0	SG0_CLK	MIIM1_MDC	SI_SLV_CLK	-
GPIO_8	RGMII1_RXC-REFCLK	I2C0_SCL6	SD1	SG0_DO	MIIM1_MDIO	-	-
GPIO_9	RGMII1_RXCTL-CRS_DV	I2C0_SCL7	RCLK0	SG0_DI	IRQ1	UART_RXD	-
GPIO_10	RGMII1_TXC-REFCLK	I2C0_SCL8	RCLK1	SG0_LD	IRQ2	UART_TXD	-
GPIO_11	RGMII1_TXCTL-TX_DV	I2C0_SCL9	MIIM1_MDC	-	IRQ3	-	-
GPIO_12	RGMII1_TXD0	I2C0_SCL10	MIIM1_MDIO	PTP0	-	-	-
GPIO_13	RGMII1_TXD1	I2C0_SCL11	-	PTP1	MAC_LED6	-	-
GPIO_14	RGMII1_TXD2	I2C0_SCL12	-	PTP2	MAC_LED5	-	-
GPIO_15	RGMII1_TXD3	I2C0_SCL13	-	PTP3	-	-	-

TABLE 3-4: GPIO ALTERNATE FUNCTION ASSIGNMENTS IN MANAGED MODE (CONTINUED)

GPIO	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	Manual
GPIO_16	RGMII0_RXD0	-	-	-	-	-	-
GPIO_17	RGMII0_RXD1	-	-	-	-	-	-
GPIO_18	RGMII0_RXD2-RXER	-	-	-	-	-	-
GPIO_19	RGMII0_RXD3	-	-	-	-	-	-
GPIO_20	RGMII0_RXC-REFCLK	-	-	-	-	-	-
GPIO_21	RGMII0_RXCTL-CRS_DV	-	-	-	-	-	-
GPIO_22	RGMII0_TXC-REFCLK	-	-	-	-	-	-
GPIO_23	RGMII0_TXCTL-TX_DV	-	-	-	-	-	-
GPIO_24	RGMII0_TXD0	-	-	-	-	-	-
GPIO_25	RGMII0_TXD1	-	-	-	-	-	-
GPIO_26	RGMII0_TXD2	-	-	-	-	-	-
GPIO_27	RGMII0_TXD3	-	-	-	-	-	-
GPIO_28	RCLK0	MIIM0_MDC	-	-	-	-	-
GPIO_29	RCLK1	MIIM0_MDIO	-	-	-	-	-
GPIO_30	PTP0	I2C0_SDA	UART_RXD	-	-	-	-
GPIO_31	PTP1	I2C0_SCL0	UART_TXD	-	-	-	-
GPIO_32	PTP2	I2C0_SCL1	-	-	-	-	-
GPIO_33	PTP3	IRQ0	-	-	-	-	-
GPIO_34	RCLK0	PHY4_LED0	PHY4_LED0	-	-	-	-
GPIO_35	RCLK1	PHY4_LED1	PHY3_LED0	-	MAC_LED0	-	-
GPIO_36	PTP0	PHY3_LED0	PHY2_LED0	-	MAC_LED1	-	-
GPIO_37	PTP1	PHY3_LED1	PHY1_LED0	-	MAC_LED2	-	-
GPIO_38	-	PHY2_LED0	PHY0_LED0	-	MAC_LED3	-	-
GPIO_39	UART_TXD	PHY2_LED1	-	-	MAC_LED4	-	TDI
GPIO_40	SI_DI	PHY1_LED0	SG0_CLK	-	MAC_LED5	-	TCK
GPIO_41	SI_DO	PHY1_LED1	SG0_DO	IRQ1	MAC_LED6	-	TDO
GPIO_42	SI_nCS	PHY0_LED0	SG0_DI	IRQ2	MAC_LED7	SD0	NTRST
GPIO_43	SI_CLK	PHY0_LED1	SG0_LD	IRQ3	MAC_LED8	SD1	TMS
EGPIO_ [50:44] - 156-Pin VQFN-DR Package Only							
EGPIO_44	MIIM1_MDC	I2C1_SDA	-	-	-	-	-
EGPIO_45	MIIM1_MDIO	I2C1_SCL	-	-	-	-	-
EGPIO_46	-	PHY2_LED1	-	-	-	-	-
EGPIO_47	-	PHY1_LED0	-	-	-	-	-

TABLE 3-4: GPIO ALTERNATE FUNCTION ASSIGNMENTS IN MANAGED MODE (CONTINUED)

GPIO	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	Manual
EGPIO_48	MIIM_SLV_ADDR	PHY1_LED1	-	-	-	-	-
EGPIO_49	MIIM_SLV_MDC	PHY0_LED0	I2C_SLV_SDA	-	-	-	-
EGPIO_50	MIIM_SLV_MDIO	PHY0_LED1	I2C_SLV_SCL	-	-	-	-

3.3.2 GPIO ALTERNATE FUNCTIONS IN UNMANAGED MODE

In unmanaged mode, GPIO alternate functions are determined based on a combination of GPIO mode and boot mode configuration strap settings. GPIO Mode Selector (GPIO_MODE[1:0]) selects between the GPIO Mode A, B, C, or D. Boot Mode Selector (BOOT_MODE[2:0]) configures the internal controller's interface mode. Available functions in each GPIO Mode are shown in Table 3-5, Table 3-6, Table 3-7, and Table 3-8. Refer to Section 3.4, Configuration Straps for further information on configuration strapping pins. Refer to Section 4.1, Unmanaged Mode for further information on unmanaged mode features.

Note: All unmanaged mode GPIO selections feature a output GPIO pin called System Ready (SYS_RDY). This GPIO signals that the internal controller has completed the initial boot and system configurations, and is now monitoring the status of the ports.

Note: **EGPIO_[55:44]** are only available in the 156-Pin VQFN-DR package, respectively.

TABLE 3-5: GPIO ALTERNATE FUNCTION ASSIGNMENTS UNMANAGED MODE A

GPIO	Boot Mode			
	ROM with I^2C Table	ROM with SPI Table	ROM without VRAP	Alternative ROM with I^2C Table (Note 3-1)
GPIO_0	UART_RXD	SI_DI	UART_RXD	UART_RXD
GPIO_1	UART_TXD	SI_DO	UART_TXD	UART_TXD
GPIO_2	I2C1_SDA	SI_nCS	-	I2C1_SDA
GPIO_3	I2C1_SCL	SI_CLK	-	I2C1_SCL
GPIO_4	RGMII1_RXD0	RGMII1_RXD0	RGMII1_RXD0	RGMII1_RXD0
GPIO_5	RGMII1_RXD1	RGMII1_RXD1	RGMII1_RXD1	RGMII1_RXD1
GPIO_6	RGMII1_RXD2-RXER	RGMII1_RXD2-RXER	RGMII1_RXD2-RXER	RGMII1_RXD2-RXER
GPIO_7	RGMII1_RXD3	RGMII1_RXD3	RGMII1_RXD3	RGMII1_RXD3
GPIO_8	RGMII1_RXC-REFCLK	RGMII1_RXC-REFCLK	RGMII1_RXC-REFCLK	RGMII1_RXC-REFCLK
GPIO_9	RGMII1_RXCTL-CRS_DV	RGMII1_RXCTL-CRS_DV	RGMII1_RXCTL-CRS_DV	RGMII1_RXCTL-CRS_DV
GPIO_10	RGMII1_TXC-REFCLK	RGMII1_TXC-REFCLK	RGMII1_TXC-REFCLK	RGMII1_TXC-REFCLK
GPIO_11	RGMII1_TXCTL-TX_DV	RGMII1_TXCTL-TX_DV	RGMII1_TXCTL-TX_DV	RGMII1_TXCTL-TX_DV
GPIO_12	RGMII1_TXD0	RGMII1_TXD0	RGMII1_TXD0	RGMII1_TXD0
GPIO_13	RGMII1_TXD1	RGMII1_TXD1	RGMII1_TXD1	RGMII1_TXD1
GPIO_14	RGMII1_TXD2	RGMII1_TXD2	RGMII1_TXD2	RGMII1_TXD2
GPIO_15	RGMII1_TXD3	RGMII1_TXD3	RGMII1_TXD3	RGMII1_TXD3
GPIO_16	RGMII0_RXD0	RGMII0_RXD0	RGMII0_RXD0	RGMII0_RXD0
GPIO_17	RGMII0_RXD1	RGMII0_RXD1	RGMII0_RXD1	RGMII0_RXD1
GPIO_18	RGMII0_RXD2-RXER	RGMII0_RXD2-RXER	RGMII0_RXD2-RXER	RGMII0_RXD2-RXER
GPIO_19	RGMII0_RXD3	RGMII0_RXD3	RGMII0_RXD3	RGMII0_RXD3

TABLE 3-5: GPIO ALTERNATE FUNCTION ASSIGNMENTS UNMANAGED MODE A

GPIO_20	RGMII0_RXC-REFCLK	RGMII0_RXC-REFCLK	RGMII0_RXC-REFCLK	RGMII0_RXC-REFCLK
GPIO_21	RGMII0_RXCTL-CRS_DV	RGMII0_RXCTL-CRS_DV	RGMII0_RXCTL-CRS_DV	RGMII0_RXCTL-CRS_DV
GPIO_22	RGMII0_TXC-REFCLK	RGMII0_TXC-REFCLK	RGMII0_TXC-REFCLK	RGMII0_TXC-REFCLK
GPIO_23	RGMII0_TXCTL-TX_DV	RGMII0_TXCTL-TX_DV	RGMII0_TXCTL-TX_DV	RGMII0_TXCTL-TX_DV
GPIO_24	RGMII0_TXD0	RGMII0_TXD0	RGMII0_TXD0	RGMII0_TXD0
GPIO_25	RGMII0_TXD1	RGMII0_TXD1	RGMII0_TXD1	RGMII0_TXD1
GPIO_26	RGMII0_TXD2	RGMII0_TXD2	RGMII0_TXD2	RGMII0_TXD2
GPIO_27	RGMII0_TXD3	RGMII0_TXD3	RGMII0_TXD3	RGMII0_TXD3
GPIO_28	MIIM0_MDC	MIIM0_MDC	MIIM0_MDC	MIIM0_MDC
GPIO_29	MIIM0_MDIO	MIIM0_MDIO	MIIM0_MDIO	MIIM0_MDIO
GPIO_30	I2C0_SDA	I2C0_SDA	I2C0_SDA	I2C0_SDA
GPIO_31	I2C0_SCL0	I2C0_SCL0	I2C0_SCL0	I2C0_SCL0
GPIO_32	I2C0_SCL1	I2C0_SCL1	I2C0_SCL1	I2C0_SCL1
GPIO_33	SYS_RDY	SYS_RDY	SYS_RDY	SYS_RDY
GPIO_34	PHY4_LED0	PHY4_LED0	PHY4_LED0	PHY4_LED0
GPIO_35	Note 3-2	Note 3-2	Note 3-2	Note 3-2
GPIO_36	Note 3-3	Note 3-3	Note 3-3	Note 3-3
GPIO_37	Note 3-4	Note 3-4	Note 3-4	Note 3-4
GPIO_38	Note 3-5	Note 3-5	Note 3-5	Note 3-5
GPIO_39	-	UART_TXD	-	-
GPIO_40	SG0_CLK	SG0_CLK	SG0_CLK	MAC_LED5
GPIO_41	SG0_DO	SG0_DO	SG0_DO	MAC_LED6
GPIO_42	SG0_DI	SG0_DI	SG0_DI	SD0
GPIO_43	SG0_LD	SG0_LD	SG0_LD	SD1
EGPIO_ [50:44] - 156-Pin VQFN-DR Package Only				
EGPIO_44	-	-	-	-
EGPIO_45	-	-	-	-
EGPIO_46	PHY2_LED1	PHY2_LED1	-	PHY2_LED1
EGPIO_47	PHY1_LED0	PHY1_LED0	-	PHY1_LED0
EGPIO_48	PHY1_LED1	PHY1_LED1	-	PHY1_LED1
EGPIO_49	PHY0_LED0	PHY0_LED0	-	PHY0_LED0
EGPIO_50	PHY0_LED1	PHY0_LED1	-	PHY0_LED1

Note 3-1 For GPIO Mode A, a direct GPIO connection to the SPFs may be used as an alternate configuration in place of using the SGPIO interface. This alternative Boot Mode A configuration is shown applied to the ROM with the I^2C Table boot mode selection column in Table 3-5, but it may be similarly applied to the other two Boot Mode A options.

Note 3-2 128-Pin TQFP = PHY3_LED0
156-Pin VQFN-DR = PHY4_LED1

Note 3-3 128-Pin TQFP = PHY2_LED0
156-Pin VQFN-DR = PHY3_LED0

Note 3-4 128-Pin TQFP = PHY1_LED0
156-Pin VQFN-DR = PHY3_LED1

Note 3-5 128-Pin TQFP = PHY0_LED0
156-Pin VQFN-DR = PHY2_LED0

TABLE 3-6: GPIO ALTERNATE FUNCTION ASSIGNMENTS UNMANAGED MODE B

GPIO	Boot Mode		
	ROM with I^2C table	ROM with SPI table	ROM without VRAP
GPIO_0	-	SI_DI	-
GPIO_1	-	SI_DO	-
GPIO_2	I2C1_SDA	SI_nCS	-
GPIO_3	I2C1_SCL	SI_CLK	-
GPIO_[15:4]	-	-	-
GPIO_16	RGMII0_RXD0	RGMII0_RXD0	RGMII0_RXD0
GPIO_17	RGMII0_RXD1	RGMII0_RXD1	RGMII0_RXD1
GPIO_18	RGMII0_RXD2-RXER	RGMII0_RXD2-RXER	RGMII0_RXD2-RXER
GPIO_19	RGMII0_RXD3	RGMII0_RXD3	RGMII0_RXD3
GPIO_20	RGMII0_RXC-REFCLK	RGMII0_RXC-REFCLK	RGMII0_RXC-REFCLK
GPIO_21	RGMII0_RXCTL-CRS_DV	RGMII0_RXCTL-CRS_DV	RGMII0_RXCTL-CRS_DV
GPIO_22	RGMII0_TXC-REFCLK	RGMII0_TXC-REFCLK	RGMII0_TXC-REFCLK
GPIO_23	RGMII0_TXCTL-TX_DV	RGMII0_TXCTL-TX_DV	RGMII0_TXCTL-TX_DV
GPIO_24	RGMII0_TXD0	RGMII0_TXD0	RGMII0_TXD0
GPIO_25	RGMII0_TXD1	RGMII0_TXD1	RGMII0_TXD1
GPIO_26	RGMII0_TXD2	RGMII0_TXD2	RGMII0_TXD2
GPIO_27	RGMII0_TXD3	RGMII0_TXD3	RGMII0_TXD3
GPIO_28	MIIM0_MDC	MIIM0_MDC	MIIM0_MDC
GPIO_29	MIIM0_MDIO	MIIM0_MDIO	MIIM0_MDIO
GPIO_30	UART_RXD	UART_RXD	UART_RXD
GPIO_31	UART_TXD	UART_TXD	UART_TXD
GPIO_32	-	-	-
GPIO_33	SYS_RDY	SYS_RDY	SYS_RDY
GPIO_34	PHY4_LED0	PHY4_LED0	PHY4_LED0
GPIO_35	PHY4_LED1	PHY4_LED1	PHY4_LED1
GPIO_36	PHY3_LED0	PHY3_LED0	PHY3_LED0
GPIO_37	PHY3_LED1	PHY3_LED1	PHY3_LED1
GPIO_38	PHY2_LED0	PHY2_LED0	PHY2_LED0
GPIO_39	PHY2_LED1	PHY2_LED1	PHY2_LED1
GPIO_40	PHY1_LED0	PHY1_LED0	PHY1_LED0
GPIO_41	PHY1_LED1	PHY1_LED1	PHY1_LED1
GPIO_42	PHY0_LED0	PHY0_LED0	PHY0_LED0
GPIO_43	PHY0_LED1	PHY0_LED1	PHY0_LED1
EGPIO_ [50:44] - 156-Pin VQFN-DR Package Only			
EGPIO_[50:44]	-	-	-

TABLE 3-7: GPIO ALTERNATE FUNCTION ASSIGNMENTS UNMANAGED MODE C

GPIO	Boot Mode		
	ROM with I^2C table	ROM with SPI table	ROM without VRAP
GPIO_0	-	SI_DI	-
GPIO_1	-	SI_DO	-
GPIO_2	I2C1_SDA	SI_nCS	-
GPIO_3	I2C1_SCL	SI_CLK	-
GPIO_4	RGMII1_RXD0	RGMII1_RXD0	RGMII1_RXD0
GPIO_5	RGMII1_RXD1	RGMII1_RXD1	RGMII1_RXD1
GPIO_6	RGMII1_RXD2-RXER	RGMII1_RXD2-RXER	RGMII1_RXD2-RXER
GPIO_7	RGMII1_RXD3	RGMII1_RXD3	RGMII1_RXD3
GPIO_8	RGMII1_RXC-REFCLK	RGMII1_RXC-REFCLK	RGMII1_RXC-REFCLK
GPIO_9	RGMII1_RXCTL-CRS_DV	RGMII1_RXCTL-CRS_DV	RGMII1_RXCTL-CRS_DV
GPIO_10	RGMII1_TXC-REFCLK	RGMII1_TXC-REFCLK	RGMII1_TXC-REFCLK
GPIO_11	RGMII1_TXCTL-TX_DV	RGMII1_TXCTL-TX_DV	RGMII1_TXCTL-TX_DV
GPIO_12	RGMII1_TXD0	RGMII1_TXD0	RGMII1_TXD0
GPIO_13	RGMII1_TXD1	RGMII1_TXD1	RGMII1_TXD1
GPIO_14	RGMII1_TXD2	RGMII1_TXD2	RGMII1_TXD2
GPIO_15	RGMII1_TXD3	RGMII1_TXD3	RGMII1_TXD3
GPIO_16	RGMII0_RXD0	RGMII0_RXD0	RGMII0_RXD0
GPIO_17	RGMII0_RXD1	RGMII0_RXD1	RGMII0_RXD1
GPIO_18	RGMII0_RXD2-RXER	RGMII0_RXD2-RXER	RGMII0_RXD2-RXER
GPIO_19	RGMII0_RXD3	RGMII0_RXD3	RGMII0_RXD3
GPIO_20	RGMII0_RXC-REFCLK	RGMII0_RXC-REFCLK	RGMII0_RXC-REFCLK
GPIO_21	RGMII0_RXCTL-CRS_DV	RGMII0_RXCTL-CRS_DV	RGMII0_RXCTL-CRS_DV
GPIO_22	RGMII0_TXC-REFCLK	RGMII0_TXC-REFCLK	RGMII0_TXC-REFCLK
GPIO_23	RGMII0_TXCTL-TX_DV	RGMII0_TXCTL-TX_DV	RGMII0_TXCTL-TX_DV
GPIO_24	RGMII0_TXD0	RGMII0_TXD0	RGMII0_TXD0
GPIO_25	RGMII0_TXD1	RGMII0_TXD1	RGMII0_TXD1
GPIO_26	RGMII0_TXD2	RGMII0_TXD2	RGMII0_TXD2
GPIO_27	RGMII0_TXD3	RGMII0_TXD3	RGMII0_TXD3
GPIO_28	MIIM0_MDC	MIIM0_MDC	MIIM0_MDC
GPIO_29	MIIM0_MDIO	MIIM0_MDIO	MIIM0_MDIO
GPIO_30	UART_RXD	UART_RXD	UART_RXD
GPIO_31	UART_TXD	UART_TXD	UART_TXD
GPIO_32	-	-	-
GPIO_33	SYS_RDY	SYS_RDY	SYS_RDY
GPIO_34	PHY4_LED0	PHY4_LED0	PHY4_LED0
GPIO_35	PHY4_LED1	PHY4_LED1	PHY4_LED1
GPIO_36	PHY3_LED0	PHY3_LED0	PHY3_LED0
GPIO_37	PHY3_LED1	PHY3_LED1	PHY3_LED1

TABLE 3-7: GPIO ALTERNATE FUNCTION ASSIGNMENTS UNMANAGED MODE C

GPIO_38	PHY2_LED0	PHY2_LED0	PHY2_LED0
GPIO_39	PHY2_LED1	PHY2_LED1	PHY2_LED1
GPIO_40	PHY1_LED0	PHY1_LED0	PHY1_LED0
GPIO_41	PHY1_LED1	PHY1_LED1	PHY1_LED1
GPIO_42	PHY0_LED0	PHY0_LED0	PHY0_LED0
GPIO_43	PHY0_LED1	PHY0_LED1	PHY0_LED1
EGPIO_ [50:44] - 156-Pin VQFN-DR Package Only			
EGPIO_[50:44]	-	-	-

TABLE 3-8: GPIO ALTERNATE FUNCTION ASSIGNMENTS UNMANAGED MODE D

GPIO	**Boot Mode**		
	ROM with I^2C table	**ROM with SPI table**	**ROM without VRAP**
GPIO_0	-	SI_DI	-
GPIO_1	-	SI_DO	-
GPIO_2	I2C1_SDA	SI_nCS	-
GPIO_3	I2C1_SCL	SI_CLK	-
GPIO_4	SFP_MOD_DET_1	SFP_MOD_DET_1	SFP_MOD_DET_1
GPIO_5	SFP_MOD_DET_0	SFP_MOD_DET_0	SFP_MOD_DET_0
GPIO_6	SFP_TX_DIS_1	SFP_TX_DIS_1	SFP_TX_DIS_1
GPIO_7	SD0	SD0	SD0
GPIO_8	SD1	SD1	SD1
GPIO_9	UART_RXD	UART_RXD	UART_RXD
GPIO_10	UART_TXD	UART_TXD	UART_TXD
GPIO_11	SFP_TX_DIS_0	SFP_TX_DIS_0	SFP_TX_DIS_0
GPIO_12	-	-	-
GPIO_13	MAC_LED6	MAC_LED6	MAC_LED6
GPIO_14	MAC_LED5	MAC_LED5	MAC_LED5
GPIO_15	-	-	-
GPIO_16	RGMII0_RXD0	RGMII0_RXD0	RGMII0_RXD0
GPIO_17	RGMII0_RXD1	RGMII0_RXD1	RGMII0_RXD1
GPIO_18	RGMII0_RXD2-RXER	RGMII0_RXD2-RXER	RGMII0_RXD2-RXER
GPIO_19	RGMII0_RXD3	RGMII0_RXD3	RGMII0_RXD3
GPIO_20	RGMII0_RXC-REFCLK	RGMII0_RXC-REFCLK	RGMII0_RXC-REFCLK
GPIO_21	RGMII0_RXCTL-CRS_DV	RGMII0_RXCTL-CRS_DV	RGMII0_RXCTL-CRS_DV
GPIO_22	RGMII0_TXC-REFCLK	RGMII0_TXC-REFCLK	RGMII0_TXC-REFCLK
GPIO_23	RGMII0_TXCTL-TX_DV	RGMII0_TXCTL-TX_DV	RGMII0_TXCTL-TX_DV
GPIO_24	RGMII0_TXD0	RGMII0_TXD0	RGMII0_TXD0
GPIO_25	RGMII0_TXD1	RGMII0_TXD1	RGMII0_TXD1
GPIO_26	RGMII0_TXD2	RGMII0_TXD2	RGMII0_TXD2
GPIO_27	RGMII0_TXD3	RGMII0_TXD3	RGMII0_TXD3
GPIO_28	MIIM0_MDC	MIIM0_MDC	MIIM0_MDC

TABLE 3-8: GPIO ALTERNATE FUNCTION ASSIGNMENTS UNMANAGED MODE D

GPIO_29	MIIM0_MDIO	MIIM0_MDIO	MIIM0_MDIO
GPIO_30	I2C0_SDA	I2C0_SDA	I2C0_SDA
GPIO_31	I2C0_SCL0	I2C0_SCL0	I2C0_SCL0
GPIO_32	I2C0_SCL1	I2C0_SCL1	I2C0_SCL1
GPIO_33	SYS_RDY	SYS_RDY	SYS_RDY
GPIO_34	PHY4_LED0	PHY4_LED0	PHY4_LED0
GPIO_35	PHY4_LED1	PHY4_LED1	PHY4_LED1
GPIO_36	PHY3_LED0	PHY3_LED0	PHY3_LED0
GPIO_37	PHY3_LED1	PHY3_LED1	PHY3_LED1
GPIO_38	PHY2_LED0	PHY2_LED0	PHY2_LED0
GPIO_39	PHY2_LED1	PHY2_LED1	PHY2_LED1
GPIO_40	PHY1_LED0	PHY1_LED0	PHY1_LED0
GPIO_41	PHY1_LED1	PHY1_LED1	PHY1_LED1
GPIO_42	PHY0_LED0	PHY0_LED0	PHY0_LED0
GPIO_43	PHY0_LED1	PHY0_LED1	PHY0_LED1
EGPIO_ [50:44] - 156-Pin VQFN-DR Package Only			
EGPIO_[50:44]	-	-	-

3.3.3 GPIO ALTERNATE FUNCTION DESCRIPTIONS

Table 3-9 provides descriptions of each of the GPIO alternate functions.

TABLE 3-9: GPIO ALTERNATE FUNCTION DESCRIPTIONS

GPIO Alternate Function Name	Description
LED	
PHY[4:0]_LED[1:0]	CuPHY LED output, two LEDs per PHY
Serial GPIO	
SG0_CLK	Serial GPIO clock output
SG0_DO	Serial GPIO data output
SG0_DI	Serial GPIO data input
SG0_LD	Serial GPIO load signal
RMII/RGMII 0-1	
RGMII[1:0]_RXD0	RMII/RGMII 0/1 receive data 0 Bit 0 of the 4 data bits that are sent by the transceiver on the receive path.
RGMII[1:0]_RXD1	RMII/RGMII 0/1 receive data 1 Bit 1 of the 4 data bits that are sent by the transceiver on the receive path.

TABLE 3-9: GPIO ALTERNATE FUNCTION DESCRIPTIONS (CONTINUED)

GPIO Alternate Function Name	Description
RGMII[1:0]_RXD2-RXER	**RGMII Mode:** RGMII 0/1 receive data 2 (RXD2) Bit 2 of the 4 data bits that are sent by the transceiver on the receive path. **RMII Mode:** RMII 0/1 receive error (RXER) This signal is asserted to indicate an error was detected somewhere in the frame presently being transferred from the transceiver. **Note:** This signal is optional in RMII mode.
RGMII[1:0]_RXD3	RGMII 0/1 receive data 3 Bit 3 of the 4 data bits that are sent by the transceiver on the receive path. **Note:** This signal is not used in RMII mode.
RGMII[1:0]_RXC-REFCLK	**RGMII Mode:** RGMII 0/1 receive clock (RXC) **RMII Mode:** RMII 0/1 reference clock (REFCLK)
RGMII[1:0]_RXCTL-CRS_DV	**RGMII Mode:** RGMII 0/1 receive control (RXCTRL) Indicates both the receive data valid (RXDV) and receive error (RXER) functions per the RGMII specification. **RMII Mode:** RMII 0/1 receive data valid (CRS_DV) Indicates that recovered and decoded data is available on the RXD pins.
RGMII[1:0]_TXC-REFCLK	**RGMII Mode:** RGMII 0/1 transmit clock (TXC) **RMII Mode:** RMII 0/1 reference clock out (REFCLK)
RGMII[1:0]_TXCTL-TX_DV	**RGMII Mode:** RGMII 0/1 transmit control (TXCTRL) Indicates both the transmit data enable (TXEN) and transmit error (TXER) functions per the RGMII specification. **RMII Mode:** RMII 0/1 carrier sense / receive data valid (TX_DV) Indicates that the receive medium is non-idle in RMII mode.
RGMII[1:0]_TXD0	RMII/RGMII 0/1 transmit data 0 Bit 0 of the 4 data bits that are received by the transceiver on the transmit path.
RGMII[1:0]_TXD1	RMII/RGMII 0/1 transmit data 1 Bit 1 of the 4 data bits that are received by the transceiver on the transmit path.
RGMII[1:0]_TXD2	RGMII 0/1 transmit data 2 Bit 2 of the 4 data bits that are received by the transceiver on the transmit path. **Note:** This signal is not used in RMII mode.
RGMII[1:0]_TXD3	RGMII 0/1 transmit data 3 Bit 3 of the 4 data bits that are received by the transceiver on the transmit pat. **Note:** This signal is not used in RMII mode.
PTP Sync	
PTP[3:0]	PTP sync clock

TABLE 3-9: GPIO ALTERNATE FUNCTION DESCRIPTIONS (CONTINUED)

GPIO Alternate Function Name	Description
MIIM Client	
MIIM_SLV_MDC	MIIM media control clock
MIIM_SLV_MDIO	MIIM control data
MIIM_SLV_ADDR	MIIM client address select **Note:** When 1, the MIIM client address is “11111”. When 0, the MIIM client address is “00000”.
MIIM Con	
MIIM[1:0]_MDC	MIIM media control clock
MIIM[1:0]_MDIO	MIIM control data
IRQ Outputs	
IRQ[3:0]	IRQ outputs
Miscellaneous	
RCLK[1:0]	Recovered clock outputs
SI Controller	
SI_nCS	Active low chip select
SI_CLK	Clock input
SI_DI	Data input
SI_DO	Data output
UART	
UART_RXD	UART receive data
UART_TXD	UART transmit data
I^2C Controller	
I2C0_SDA	Two-wire serial interface data
I2C0_SCL[13:0]	Two-wire serial interface gated clocks
I2C1_SDA	Two-wire serial interface data
I2C1_SCL	Two-wire serial interface clock
I^2C Client	
I2C_SLV_SDA	Two-wire serial interface data
I2C_SLV_SCL	Two-wire serial interface clock
SI Client	
SI_SLV_nCS	Active low chip select
SI_SLV_CLK	Clock input

TABLE 3-9: GPIO ALTERNATE FUNCTION DESCRIPTIONS (CONTINUED)

GPIO Alternate Function Name	Description
SI_SLV_DI	Data input
SI_SLV_DO	Data output
Signal Detect	
SD[1:0]	Signal detect input from SFP modules
SFP_MOD_DET_[1:0]	Module detect from external SFP module Must be connected to SFPs' MOD-DEF0 pin. **Note:** Only for usage in Unmanaged Mode D. See Section 3.3.2, "GPIO Alternate Functions in Unmanaged Mode" for further details.
SFP_TX_DIS_[1:0]	Disables transmitter optical output on external SFP modules Must be connected to SFPs' TX-Disable pin. **Note:** Only for usage in Unmanaged Mode D. See Section 3.3.2, "GPIO Alternate Functions in Unmanaged Mode" for further details.
System Ready	
SYS_RDY	Signals that the internal controller has completed the initial boot and system configurations System Ready will continue to monitor the status of the ports. **Note:** Only for usage in Unmanaged modes. See Section 3.3.2, "GPIO Alternate Functions in Unmanaged Mode" for further details.
MAC LED	
MAC_LED[8:0]	LED output based on MAC status
JTAG	
TDI	JTAG (IEEE 1149.1) data input
TCK	JTAG (IEEE 1149.1) test clock
TDO	JTAG (IEEE 1149.1) test data output
NTRST	JTAG (IEEE 1149.1) TAP controller reset **Note:** When JTAG is not used, tie this pin to ground or pull low
TMS	JTAG (IEEE 1149.1) test mode select

3.4 Configuration Straps

Configuration straps allow various features of the device to be automatically configured to user defined values. Configuration straps are latched upon the release of pin reset (**nRESET**) and combined with any strap override register settings at chip top level. The results are then passed on to the functions using these straps. All strap signals have an internal pull-up. However, it is recommended to always connect an external pull-up/pull-down.

Note:	The system designer must ensure that configuration strap pins meet timing requirements. The system designer must guarantee that configuration strap pins meet the timing requirements specified in Section 5.6.3, nRESET Timing Specifications. If configuration strap pins are not at the correct voltage level prior to being latched, the device may capture incorrect strap values.

Note:	When externally pulling configuration straps high, the strap must be tied to the associated **VDD_IO_*x*** rail.

APPLICATION NOTE: All straps must be pulled-up or pulled-down externally on the PCB to enable the desired operational state.

LAN9645xF has 16 features that are supported via 22 configuration strapping pins. **BOOT_MODE[2:0]**, **MGMT_MODE[1:0]**, **JTAG_EN**, and **SD_125M** are applicable in both managed or unmanaged modes. All of the remaining strapping pins are used in unmanaged mode only. Refer to Section 4.1, Unmanaged Mode for additional information on unmanaged mode features.

3.4.1 MANAGED/UNMANAGED MODE CONFIGURATION STRAPS

3.4.1.1 Boot Mode Selector (BOOT_MODE[2:0])

These configuration straps select the booting interface for the internal controller. Unmanaged modes are selected using options **3**, **4**, **5**, **6**, or **7**.

- 0: No boot. The internal controller does not boot; however it is possible to load executable code into the internal RAM using a management interface to manually start the internal controller, releasing its reset.
- 1: RESERVED
- 2: RESERVED
- 3: ROM boot. The internal controller executes from the internal ROM.
- 4: ROM + I^2C table boot. The internal controller executes from the internal ROM, the I^2C_1 interface is exposed on the GPIOs, and the internal controller reads the EEPROM configuration table. The **EEPROM_SIZE** strap addressing mode must be set accordingly.
- 5: ROM + SPI table boot. The internal controller executes from the internal ROM, the SPI host interface is exposed on the GPIOs, and the internal controller reads the EEPROM configuration table.
- 6: ROM + VRAP port 5 boot. The internal controller executes from the internal ROM but no additional interface is exposed on the GPIOs (port 5 is enabled to receive VRAP frames to access registers). See Section 4.19, VRAP Engine for details.
- 7: ROM + VRAP port 7 boot. The internal controller executes from the internal ROM but no additional interface is exposed on the GPIOs (port 7 is enabled to receive VRAP frames to access registers). See Section 4.19, VRAP Engine for details.

See Section 3.4.1.3, JTAG Enable (JTAG_EN) for additional **BOOT_MODE0** parameters.

Note:	These configuration straps apply for both managed and unmanaged mode.

3.4.1.2 Management Interface Mode Selector (MGMT_MODE[1:0])

This configuration strap selects which management interfaces to expose on GPIOs:

- 0: UART CLI
- 1: I^2C client
- 2: MIIM client
- 3: SPI client

The UART interface is exposed on GPIOs when the internal controller is booting from internal ROM.

For other interfaces, alternate function pin **UART_TXD** is enabled in GPIO mode A operation, but only for printing logging messages. See Section 3.4.2.9, GPIO Mode Selector (GPIO_MODE[1:0]) for further details.

Note: These configuration straps apply for both managed and unmanaged mode. UART is only supported in unmanaged mode.

3.4.1.3 JTAG Enable (JTAG_EN)

This configuration strap enables the JTAG interface.

- 0: Disable JTAG
- 1: Enable JTAG

If JTAG is enabled, configuration strap **BOOT_MODE0** selects:

- 0: DFT TAP controller
- 1: SerDes TAP controller

Note: This configuration strap applies for both managed and unmanaged mode.

3.4.1.4 SerDes 125 MHz Selector (SD_125M)

This configuration strap enables a 125 MHz reference clock input to SerDes. The 125 MHz clock may be used in combination with the QSGMII to reduce jitter on the SerDes pins (instead of a 25 MHz clock signal from the PLL).

Note: If **SD_125M** is strapped high, **SGPIO_EN** must be strapped low. See Section 3.4.2.10, SGPIO Selector (SGPIO_EN) for further details.

Note: This configuration strap applies for both managed and unmanaged mode.

3.4.2 UNMANAGED MODE CONFIGURATION STRAPS

3.4.2.1 Loop Prevention Enable (LOOP_EN)

Configuration strap **LOOP_EN** enables network loop detection and prevention. By monitoring the MAC table learn activity, the internal controller can detect a potential loop in the network if a given MAC address is continuously learned on different ports. If a loop is detected, the involved ports are disabled for a period of 10 seconds. This prevents the malfunction caused by the network loop. The ports are then re-enabled after the 10 seconds have elapsed. If the loop persists, the loop detection will once again disable the ports.

- 0: Disable loop prevention
- 1: Enable loop prevention

Note: This configuration strap is applicable to unmanaged mode only.

3.4.2.2 End-to-end Transparent Clock Enable (E2E_CLOCK_EN)

Configuration strap **E2E_CLOCK_EN** enables a stateless one-step, end-to-end transparent clock where the correction field in Sync and Delay_Req PTP frames are updated with the observed residence delay for PTP frames passing through the switch. The feature supports the following PTP frame encapsulations:

- PTP over Ethernet (EtherType = 0x88F7)
- PTP over UDP over IPv4 (UDP destination port number 319)
- PTP over UDP over IPv6 (UDP destination port number 319)
- PTP message types of Sync and Delay_Req are processed.

The **E2E_CLOCK_EN** settings are:

- 0: Disable end-to-end transparent clock
- 1: Enable end-to-end transparent clock

Note: This configuration strap is applicable to unmanaged mode only.

3.4.2.3 Flow Control Enable (FLOW_EN)

Configuration strap **FLOW_EN** enables flow control advertisement on all applicable ports. Whether flow control is enabled on a link also depends on the link partner.

Pause frames generated by the switch use source MAC address 30:05:02:C1:01:0x, where x is the port index. For example, port 4 uses source MAC address 30:05:02:C1:01:04.

The MAC address can be changed by using an EEPROM/flash configuration table.

- 0: Disable flow control
- 1: Enable flow control

Note: This configuration strap is applicable to unmanaged mode only.

3.4.2.4 Energy Efficient Ethernet Enable (EEE_EN)

This configuration strap enables Energy Efficient Ethernet on CuPHY ports.

- 0: Disable EEE on CuPHY ports
- 1: Enable EEE on CuPHY ports

Note: This configuration strap is applicable to unmanaged mode only.

3.4.2.5 LED Mode Selector (LED_MODE)

Multiple LED behaviors are supported depending on I/O configuration and package.

If only one LED per port is available for the GPIO mode selected, the LED is active (active low) when link-up and blinking with link activity.

If two LEDs per port are available, it is possible to select between two behaviors with configuration strap **LED_MODE**.

- 0: Single color mode (one LED signaling link-up and one led signaling activity)
- 1: Tri-color mode (two LEDs to indicate link-up with different color depending on link speed, LEDs blinking with activity)

Note: This configuration strap is applicable to unmanaged mode only.

3.4.2.6 CuPHY Mode Selector (CUPHY_MODE)

This configuration strap selects if 1G speed should be advertised for integrated CuPHYs. Advertising for external CuPHYs will not be changed.

- 0: Auto negotiation, Auto MDI-X – all advertised
- 1: Auto negotiation, Auto MDI-X – 100FD/HD, 10FD/HD advertised

Note: This configuration strap is applicable to unmanaged mode only.

3.4.2.7 2.5G SFP Speed Detect Enable (SFP_EN)

This configuration strap is only applicable if SFPs are used.

When enabled, if a cable is plugged in but a 2.5G SFP link-up is not detected, the device will attempt to link at lower speeds (1000BASE-X for optical SFPs and SGMII for copper SFPs).

If it is impossible to get a link at lower speed, the speed will be reset at 2.5G and the cycle will continue until either link is gained or the cable is disconnected.

- 0: Disable 2.5G detect
- 1: Enable 2.5G detect

Note: This configuration strap is applicable to unmanaged mode only.

3.4.2.8 EEPROM Size Selector (EEPROM_SIZE)

This configuration strap is only applicable in ROM + I^2C table boot mode. See Section 3.4.1.1, Boot Mode Selector (BOOT_MODE[2:0]) for further details.

Selects the addressing size of the I^2C EEPROM. It is not required to configure this strap if the EEPROM is only used for booting or if no I^2C EEPROM is used.

- 0: 16-bit addressing
- 1: 8-bit addressing

Note: This configuration strap is applicable to unmanaged mode only.

3.4.2.9 GPIO Mode Selector (GPIO_MODE[1:0])

These configuration straps select between four Ethernet port interface modes, as shown in Table 3-10. The GPIO mode (A, B, C, or D) also determines the mapping of GPIO alternate functions, as detailed in Section 3.3.2, GPIO Alternate Functions in Unmanaged Mode.

- 0: Mode A (2x RGMII, 1-2x SGMII)
- 1: Mode B (1x QSGMII, 0x SGMII, 0-1x RGMII)
- 2: Mode C (2x RGMII, 0x SGMII)
- 3: Mode D (0-1x RGMII, 0-2x SGMII

Note: This configuration strap is applicable to unmanaged mode only.

TABLE 3-10: GPIO MODE INTERFACE OPTIONS

GPIO Mode	RGMII Interfaces	SGMII Interfaces	QSGMII Support
A	2	1-2	No
B	0-1	0	Yes
C	2	0	No
D	0-1	0-2	No
Note: The number of CuPHYs does not influence the GPIO mode.			

Note: This configuration strap is applicable to unmanaged mode only.

3.4.2.10 SGPIO Selector (SGPIO_EN)

In GPIO Mode A, this configuration strap controls the SFP signals that can be connected, using either SGPIOs or regular GPIOs. For GPIO Modes B, C, or D, **SGPIO_EN** is unused. See Section 3.4.2.9, GPIO Mode Selector (GPIO_MODE[1:0]) for further details.

- 0: GPIOs will be used to control SFP signals
- 1: SGPIOS will be used to control SFP signals

Note: This configuration strap is applicable to unmanaged mode only.

3.4.2.11 SerDes Interface Mode SELECTOR (SD_MODE[1:0])

These configuration straps are only applicable in GPIO modes A and D. See Section 3.4.2.9, GPIO Mode Selector (GPIO_MODE[1:0]) for further details.

Selects if SerDes 1-0 should operate in SGMII mode (for usage of external SGMII CuPHYs accessed through MIIM) or in SFP mode (for usage of SFPs accessed using I^2C). Each **SD_MODE[1:0]** strap number controls a corresponding SerDes interface number (SerDes 1-0).

- 0: SGMII (1G, 2.5G)
- 1: SFP (-X, -KX, -FX)

Note: For SFP, reading EEPROM in SFP module determines SFP mode and speed.

Note: These configuration straps are applicable to unmanaged mode only.

3.4.2.12 RGMII/RMII Interface Mode Selector (XMII_MODE[1:0])

These configuration straps select the RGMII/RMII interface mode. Each strap number controls a corresponding RGMII interface number. RGMII interface selection operates differently depending on the GPIO mode selected. See Section 3.4.2.9, GPIO Mode Selector (GPIO_MODE[1:0]) for further details.

GPIO modes A, C, and D:

XMII_MODE0 - RGMII/RMII interface mode for RGMII #0

- 0: RGMII
- 1: RMII

XMII_MODE1 - RGMII/RMII interface mode for RGMII #1

- 0: RGMII
- 1: RMII

GPIO mode B:

XMII_MODE0 - RGMII/RMII interface mode for RGMII #0

- 0: RGMII
- 1: RMII

XMII_MODE1 - RGMII mux enabling for RGMII #1

- 0: RGMII enabled instead of CuPHY #4
- 1: CuPHY #4 enabled

Note:	These configuration straps are applicable to unmanaged mode only.

4.0 FUNCTIONAL DESCRIPTIONS

This section provides information about the functional aspects of the LAN9645xF TSN Gigabit Ethernet switch device, available configurations, operational features, and testing functionality.

4.1 Unmanaged Mode

LAN9645xF features an unmanaged mode of operation, which enables an internal controller to autonomously configure switch and Ethernet ports, while continuously monitoring each port's link status.

The unmanaged mode of operation supports Layer-2 Ethernet switch features such as auto-negotiation, SerDes configuration, MAC address learning and aging, flow control, Energy Efficient Ethernet, loop prevention, and an end-to-end transparent clock.

Configuration pin straps control the overall behavior of the switch, providing a way to select between different port configurations, as well as enabling or disabling specific features. See Section 3.4, Configuration Straps for pin strapping details.

Additional unmanaged mode features, such as VLAN and Quality of Service configurations, are found in an external configuration table which can be stored in an EEPROM or a flash memory, providing means to override default configurations.

4.1.1 UNMANAGED MODE GENERAL FEATURES

- Port link status monitoring, providing external PHY management
- Autolearning and aging of MAC table entries
- Four main I/Os configurations:
  - 5xCuPHYs (with one status LED per PHY) + 2xRGMII/RMII + 2xSFPs/SGMII
  - 5xCuPHYs + 1xQSGMII
  - 5xCuPHYs (with two status LEDs per PHY) + 2xRGMII/RMII
  - 5xCuPHYs (with two status LEDs per PHY) + 1xRGMII/RMII + 2xSFPs/SGMII
- LED indications for speeds (10M/100M/1G/2.5G) and link activity for every copper port and SFP
- Serial GPIO controller for the SFP module's control signals and LED control
- IEEE 1588 1-step End-to-End Transparent Clock
- IEEE 802.3az Energy-Efficient Ethernet for copper ports
- Loop prevention
- Flow control
- VRAP engine configuration
- UART command line interface
- Configuration via external EEPROM or flash memory

4.1.2 UNMANAGED MODE FUNCTIONAL DESCRIPTIONS

4.1.2.1 Unmanaged Mode of Operation Selection

Unmanaged mode is enabled by configured via the **BOOT_MODE[2:0]** configuration straps. Unmanaged mode provides the following boot mode options:

- The internal controller starts up and takes control of the switch.
- The internal controller starts up, takes control of the switch, and the controller reads an external EEPROM/flash configuration table using I^2C.
- The internal controller starts up, takes control of the switch, and the controller reads an external EEPROM/flash configuration table using SPI.
- The internal controller starts up, takes control of the switch, and the controller enables VRAP on port 5 or port 7. See Section 4.19, VRAP Engine for details.

See Section 3.4.1.1, Boot Mode Selector (BOOT_MODE[2:0]) for additional **BOOT_MODE[2:0]** details.

4.1.2.2 GPIO Modes

Four GPIO modes (A, B, C, or D), controlled by the **GPIO_MODE[1:0]** configuration straps, determine the Ethernet port configuration and GPIO alternate function mapping. Interface options are listed in Table 4-1.

TABLE 4-1: GPIO MODE INTERFACE OPTIONS

GPIO Mode	RGMII Interfaces	SGMII Interfaces	QSGMII Support
A	2	1-2	No
B	0-1	0	Yes
C	2	0	No
D	0-1	0-2	No
Note: The number of CuPHYs does not influence the GPIO mode.			

See Section 3.4.2.9, GPIO Mode Selector (GPIO_MODE[1:0]) for additional **GPIO_MODE[1:0]** details and Section 3.3.2, GPIO Alternate Functions in Unmanaged Mode for full GPIO alternate function options.

4.1.2.3 RGMII/RMII Interface Mode

Select whether each RGMII interface operates in RGMII or RMII mode using the **XMII_MODE[1:0]** configuration straps.

See Section 3.4.2.12, RGMII/RMII Interface Mode Selector (XMII_MODE[1:0]) for full **XMII_MODE[1:0]** parameters.

4.1.2.4 SerDes Interface Mode

Each SerDes interface can be set to operate either in SGMII mode or SFP mode using the **SD_MODE[1:0]** configuration straps. In SGMII mode, external CuPHYs are accessed using MIIM. In SFP mode, external SFPs are accessed using I^2C.

See Section 3.4.2.11, SerDes Interface Mode SELECTOR (SD_MODE[1:0]) for full **SD_MODE[1:0]** parameters.

4.1.2.5 SFP 2.5G Speed Detect

When the **SFP_EN** configuration strap is enabled and a 2.5G SFP link-up is not detected, the device will attempt to link at lower speeds. If it is not possible to link at a lower speed, the speed will be reset to 2.5G. The cycle will continue until either a 2.5G link is established or the cable is disconnected.

See Section 3.4.2.7, 2.5G SFP Speed Detect Enable (SFP_EN) for full **SFP_EN** parameters.

4.1.2.6 Integrated Copper PHYs Speed Advertising

Disable the 1G speed advertisement for all integrated CuPHYs using the **CUPHY_MODE** configuration strap (advertising for external PHYs is not changed by this configuration).

See Section 3.4.2.6, CuPHY Mode Selector (CUPHY_MODE) for full **CUPHY_MODE** parameters.

4.1.2.7 LED Port Status Indication

The **LED_MODE** configuration strap selects between single color (one LED for link-up and one led for activity) and tri-color LED modes (two LEDs to indicate link-up, with a different color depending on link speed, and LEDs blinking with activity).

See Section 3.4.2.5, LED Mode Selector (LED_MODE) for full **LED_MODE** parameters.

4.1.2.8 IEEE 1588 One-Step End-to-End Transparent Clock

Configuration strap **E2E_CLOCK_EN** enables a stateless one-step, end-to-end transparent clock supporting the following PTP frame encapsulations:

- PTP over Ethernet
- PTP over UDP over IPv4
- PTP over UDP over IPv6

See Section 3.4.2.2, End-to-end Transparent Clock Enable (E2E_CLOCK_EN) for full **E2E_CLOCK_EN** parameters.

4.1.2.9 IEEE 802.3az Energy-Efficient Ethernet

Energy-Efficient Ethernet is supported on integrated CuPHY ports and can be enabled with configuration strap **EEE_EN**. The configuration applies to all internal CuPHY ports.

See Section 3.4.2.4, Energy Efficient Ethernet Enable (EEE_EN) for full **EEE_EN** parameters.

4.1.2.10 Loop Detection and Prevention

The **LOOP_EN** configuration strap enables loop detection. When the same SMAC address is seen on different ports, the ports involved in the loop will be disabled for 10 seconds.

See Section 3.4.2.1, Loop Prevention Enable (LOOP_EN) for full **LOOP_EN** parameters.

4.1.2.11 Flow Control

Configuration strap **FLOW_EN** enables flow control advertisement on all applicable ports. Whether flow control is enabled on a link also depends on the link partner.

See Section 3.4.2.3, Flow Control Enable (FLOW_EN) for full **FLOW_EN** parameters.

4.1.2.12 Port Status Monitoring

The controller continuously monitors the PHYs (copper and optical) to set up the switch ports based on whether the link is up or down. If the link is up, the controller monitors the current speed, duplex mode, pause capabilities, and updates the settings of the switch ports accordingly. PHYs are polled every 10 ms.

The internal controller uses a shared MIIM (**MIIM0_MDC**, **MIIM0_MDIO**) to manage and control external devices. It is mandatory that the following MIIM addresses are used:

- A device connected to RGMII#0 must use MIIM address 1
- A device connected to RGMII#1 must use MIIM address 3
- A device connected to SGMII#0 must use MIIM address 4
- A device connected to SGMII#1 must use MIIM address 6
- A device connected through Quad-SGMII to SGMII#0 must use MIIM addresses 7, 8, 9, and 10

4.1.2.13 MAC Table Auto-learning and Aging

The switch is configured to auto-learn new MAC addresses in the MAC table. The MAC table is scanned for unused entries yielding an approximate age time of 300 seconds.

4.1.2.14 UART Command Line Interface

While operating in unmanaged mode, a UART interface can provide a command line interface that supports basic commands for switch and PHY register access, as well as device debugging.

In GPIO mode A, the only UART pin available is **UART_TXD**, limiting the interface to only receive debug information from the internal controller.

4.1.2.15 Configuration Table

Unmanaged mode supports an external configuration table stored in EEPROM or flash memory that can enhance the unmanaged capabilities by overriding existing configurations or specifying new configurations.

Use of the configuration table is controlled through the Boot Mode Selector (BOOT_MODE[2:0]) configuration strap which allows selection between I^2C or SPI to read the external device.

When a configuration table is enabled, the internal controller reads and executes instructions from the external device during initialization or when a port link goes up or down. The instructions set provide access to all of the switch and PHY registers through read-modify-write register commands, as well as the default unmanaged configuration determined by the configuration straps.

The configuration table enables a range of enhanced features that can be applied to the switch during an unmanaged mode of operation. Table 4-2 lists some examples of what can be applied to the switch without requiring an external host processor.

TABLE 4-2: CONFIGURATION TABLE FEATURES

Feature	Description
GPIO configuration	Alterations to the overall configuration determined by **GPIO_MODE[1:0]**.
VLAN configuration	Configuration of ports and VLAN table, with support for VLAN-tagged frames. Capable of up to 4K VLAN configurations.

TABLE 4-2: CONFIGURATION TABLE FEATURES (CONTINUED)

System MAC address	Configuration of the default MAC address used by flow control.
PHY setup	Specific PHY configurations applied at initialization, during link-up, or link-down.
MAC table age time	Configuration of the age time in the MAC table. The default value is 300 seconds per IEEE.
Port mirroring	Configuration of Rx and/or Tx mirroring, where traffic received on a port (or transmitted on a port) is mirrored to another port (the mirror destination port).
Storm policing	Configuration of a storm policer. Polices flooded traffic to a maximum number of frames per seconds. Unicast, multicast, and broadcast are supported.
Quality of Service classification	Configuration of QoS classification on a port. Three methods can be used: 1. Default value 2. PCP/DEI from VLAN tags 3. DSCP value from IP header
Private VLANs	Configuration of ports for private VLAN membership, limiting traffic depending on the port type (isolated, promiscuous, community).
ACL setup	Firewall rule denying frames with a specific EtherType or IP protocol.
Link aggregation	Configuration of two or more ports to act as a logical port with increased bandwidth.
FRER setup	Static stream classification, FRER sequence number generation, and FRER sequence number recovery.

When using I^2C to access the configuration table, the following applies:

- The I^2C base address must be set to 0x50. Since up to 128 KB of memory can be accessed, multiple banks of EEPROM are accessed increasing the I^2C address.
- The memory must support at least 400 KHz clock frequency.
- Both 8-bit and 16-bit addressing are supported - this is selected by configuration strap **EEPROM_SIZE**

When using SPI to access the configuration table, the following applies:

- The memory must support at least 10 MHz clock frequency.
- Only 24-bit addressing is supported.

4.2 Ethernet Port Numbering and Multiplexing

The device provides 5, 7, or 9 Ethernet ports and 1 internal CPU port. The device has in total 9 Ethernet interfaces consisting of 5 copper transceivers, 2 SerDes, and 2 RGMII interfaces. One of the SerDes supports QSGMII.

The switch core contains an internal port numbering domain where ports are numbered from 0 through 9. A port connects to a port module, which connects to an interface macro, which connects to I/O pins on the device. When using RGMII, the port module connects directly to the RGMII pins without the need for an interface macro.

The following table shows the port numbering and how the ports map to port modules. In addition, the table lists which Ethernet interfaces can be supported by the port.

TABLE 4-3: PORT NUMBERING

Port Number	Port Module	Maximum Port Speed	Ethernet interface options
0	DEV0	1 Gbps	CuPHY #0
1	DEV1	1 Gbps	CuPHY #1
2	DEV2	1 Gbps	CuPHY #2
3	DEV3	1 Gbps	CuPHY #3
4	DEV4	1 Gbps	CuPHY #4 or RGMII #0

TABLE 4-3: PORT NUMBERING (CONTINUED)

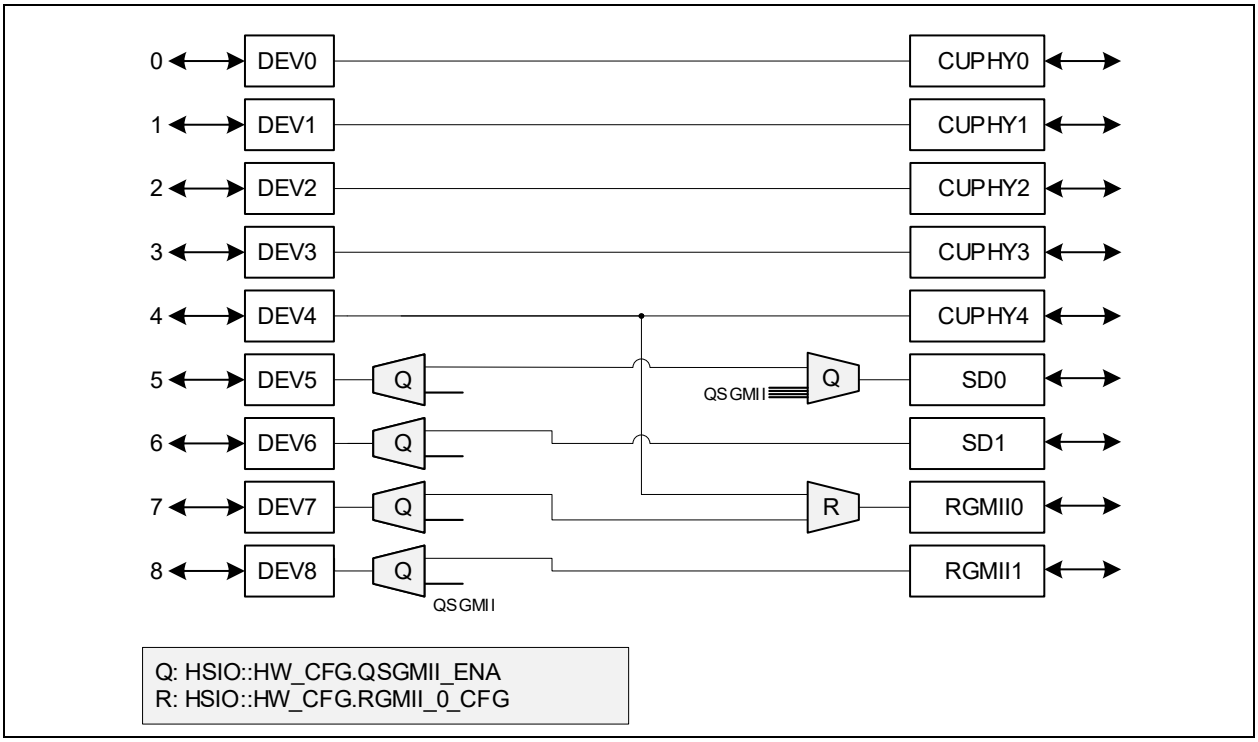
Port Number	Port Module	Maximum Port Speed	Ethernet interface options
5	DEV5	2.5 Gbps	SerDes #0
6	DEV6	2.5 Gbps	SerDes #1 or SerDes #0 (QSGMII)
7	DEV7	1 Gbps	RGMII #0 or SerDes #0 (QSGMII)
8	DEV8	1 Gbps	RGMII #1 or SerDes #0 (QSGMII)
9 (CPU)	No port module		Internal

Port 9 is an internal port used for manual injection and extraction of frames towards a CPU or processing of VRAP frames. It does not have an associated port module.

4.2.1 PORT MULTIPLEXING

The port multiplexing defines how the logical Ethernet ports connect to the Ethernet interfaces. The port multiplexing is controlled through register HSIO::HW_CFG. The following illustration highlights the available port multiplexing.

FIGURE 4-1: PORT MULTIPLEXING



As a result of the port multiplexing, the device supports a range of port configurations. Table 4-4, Table 4-5, and Table 4-6 list the active ports in major supported port configurations.

TABLE 4-4: LAN96455 PORT CONFIGURATIONS (5 PORTS)

LAN96455 Port Configuration	CuPHY					SGMII		RGMII	
	#0	#1	#2	#3	#4	#0	#1	#0	#1
5x CuPHY	DEV0	DEV1	DEV2	DEV3	DEV4	-	-	-	-
4x CuPHY + 1x 2.5G SGMII	DEV0	DEV1	DEV2	DEV3	-	DEV5	-	-	-
4x CuPHY + 1x RGMII	DEV0	DEV1	DEV2	DEV3	-	-	-	DEV7	-
3x CuPHY + 2x 2.5G SGMII	DEV0	DEV1	DEV2	-	-	DEV5	DEV6	-	-
3x CuPHY + 2x RGMII	DEV0	DEV1	DEV2	-	-	-	-	DEV7	DEV8
3x CuPHY + 1x 2.5G SGMII + 1x RGMII	DEV0	DEV1	DEV2	-	-	DEV5	-	DEV7	-
2x CuPHY + 1x 2.5G SGMII + 2x RGMII	DEV0	DEV1	-	-	-	DEV5	-	DEV7	DEV8
2x CuPHY + 2x 2.5G SGMII + 1x RGMII	DEV0	DEV1	-	-	-	DEV5	DEV6	DEV7	-
1x CuPHY + 2x 2.5G SGMII + 2x RGMII	DEV0	-	-	-	-	DEV5	DEV6	DEV7	DEV8

TABLE 4-5: LAN96457 PORT CONFIGURATIONS (7 PORTS)

LAN96457 Port Configuration	CuPHY					SGMII		RGMII	
	#0	#1	#2	#3	#4	#0	#1	#0	#1
5x CuPHY + 2x 2.5G SGMII	DEV0	DEV1	DEV2	DEV3	DEV4	DEV5	DEV6	-	-
5x CuPHY + 2x RGMII	DEV0	DEV1	DEV2	DEV3	DEV4	-	-	DEV7	DEV8
5x CuPHY + 1x 2.5G SGMII + 1x RGMII	DEV0	DEV1	DEV2	DEV3	DEV4	DEV5	-	DEV7	-
4x CuPHY + 1x 2.5G SGMII + 2x RGMII	DEV0	DEV1	DEV2	DEV3	-	DEV5	-	DEV7	DEV8
4x CuPHY + 2x 2.5G SGMII + 1x RGMII	DEV0	DEV1	DEV2	DEV3	-	DEV5	DEV6	DEV7	-
3x CuPHY + 2x 2.5G SGMII + 2x RGMII	DEV0	DEV1	DEV2	-	-	DEV5	DEV6	DEV7	DEV8
3x CuPHY + 1x QSGMII	DEV0	DEV1	DEV2	-	-	DEV5 DEV6 DEV7 DEV8	-	-	-
2x CuPHY + 1x QSGMII + 1x RGMII	DEV0	DEV1	-	-	-	DEV5 DEV6 DEV7 DEV8	-	DEV4	-

TABLE 4-6: LAN96459 PORT CONFIGURATIONS (9 PORTS)

LAN96459 Port Configuration	CuPHY					SGMII		RGMII	
	#0	#1	#2	#3	#4	#0	#1	#0	#1
5x CuPHY + 2x 2.5G SGMII + 2x RGMII	DEV0	DEV1	DEV2	DEV3	DEV4	DEV5	DEV6	DEV7	DEV8
5x CuPHY + 1x QSGMII	DEV0	DEV1	DEV2	DEV3	DEV4	DEV5 DEV6 DEV7 DEV8	-	-	-
4x CuPHY + 1x QSGMII + 1x RGMII	DEV0	DEV1	DEV2	DEV3	-	DEV5 DEV6 DEV7 DEV8	-	DEV4	-

4.2.2 SUPPORTED SERDES INTERFACES

The device supports a range of SerDes interfaces. The following table lists the SerDes interfaces supported by the SerDes ports, including standards, data rates, connectors, medium, and coding for each interface.

TABLE 4-7: SUPPORTED SERDES INTERFACES

Port Interface	Specification	Port Speed	Data Rate	Connector	Medium	Coding
SGMII	Cisco	1G	1.25 Gbps		PCB	8B10B
SFP	SFP-MSA	1G	1.25 Gbps	SFP	PCB	8b10B
100BASE-FX	IEEE802.3, Clause 24	100M	125 Mbps	SFP	PCB	4B5B
1000BASE-KX	IEEE802.3, Clause 70	1G	1.25 Gbps		PCB, backplane	8B10B
2.5G	Proprietary (aligned to SFP)	2.5G	3.125 Gbps		PCB	8B10B
Quad-SGMII	Cisco	4 x 1G	5 Gbps		PCB, backplane	8B10B

4.2.3 LOGICAL PORT NUMBERS

In many places, the analyzer and the rewriter use a logical port number. For instance, when link aggregation is enabled, all ports within a link aggregation group must be configured to use the physical port number of the port with the lowest port number within the group as logical port group ID. The mapping to a logical port number is configured in ANA:PORT:PORT_CFG.PORTID_VAL.

4.3 Copper Transceivers

The device integrates five CuPHYs (copper physical layer) low-power triple-speed (10BASE-T/100BASE-TX/1000BASE-T) Ethernet transceivers that support transmission and reception of data on standard CAT-5, as well as CAT-5e and CAT-6, unshielded twisted pair (UTP) cables. They offer diagnostic features to facilitate system bring-up and debugging in production testing and in product deployment. The LinkMD® TDR-based cable diagnostic identifies faulty copper cabling. Integrated loopback functions verify analog and digital data paths.

The registers of the integrated transceivers are not placed in the memory map of the switch but are attached instead to the built-in MII management controller 1. As a result, PHY registers are accessed indirectly through the switch registers.

4.4 SERDES

The Ethernet SerDes interfaces support different operating modes and line rates. SerDes features include:

- Programmable RX Equalizer
- Adjustable TX de-emphasis and amplitude
- Loss of Signal Detector

- Supports RX Eye Monitor
- TX-to-RX Loopback
- Integrated PRBS generator and checker
- Polarity inversion control

4.4.1 SUPPORTED LINE RATES

The high-speed SerDes interface can operate at the following data rates:

- 125 Mbps (100BASE-FX)
- 1.25 Gbps (SGMII/1000BASE-X)
- 3.125 Gbps (2.5GBASE-X)
- 5 Gbps (QSGMII)

4.4.2 RX EQUALIZATION

The differential receiver includes an equalizer. The equalizer allows the receiver to recover data transmitted over lossy backplane channels.

4.4.3 TX EQUALIZATION

The transmitter includes a programmable equalization feature to compensate for frequency-dependent loss in the channel. This is accomplished by configurable pre-emphasis.

The output amplitude can be configured to allow compliance with the different standards.

4.4.4 LOSS OF SIGNAL DETECTOR

The SerDes contains an internal Loss of Signal (LOS) detector that can be configured to detect if activity is present on the serial RX signals. This can be used to disqualify spurious input noise due to crosstalk for backplane connections. For optical SFP connections, it is recommended to use the external LOS signal from the SFP instead.

Note that at signaling rates above 1.25 Gbps, the LOS detector is unreliable and must be disabled.

4.4.5 RX EYE MONITOR

The receive side can be configured in a special RX Eye monitor mode that allows the user to get a representation of the receive eye after the RX equalization. This debug feature is beneficial when debugging bit errors and optimizing RX and TX equalization.

Note that the RX eye monitor mode cannot be used while receiving data in normal operation as it reconfigures the receiver circuits.

4.4.6 LOOPBACK TEST MODE

TX to RX loopback mode is supported by the SerDes. The loopback happens close to the pins on the serial signal.

4.4.7 PRBS FUNCTION

The SerDes provides an integrated PRBS generator and checker, allowing fast channel testing with standard compliant test patterns. The PRBS generator/checker supports PRBS-7 and PRBS-15 patterns.

4.4.8 POLARITY INVERSION

The SerDes support inversion of the TX and RX differential pairs to allow the PCB signal routing to be optimized to a given PCB layout.

4.5 RGMII

This section provides information about the RGMII functionality. The device includes two RGMII interfaces: RGMII_0 and RGMII_1. RGMII is a reduced GMII interface with a maximum speed of 1000 Mbps.

The RGMII interfaces can be used by a limited number of devices:

- RGMII_0: Port Module 4 or Port Module 7
- RGMII_1: Port Module 8

4.5.1 RESET AND DLL CONFIGURATION

The RGMII interfaces must be enabled and assigned to a device before they can be used:

- RGMII_0_CFG and RGMII_1_CFG: Assign RGMII to a device
- RGMII_ENA: Enable RGMII ports on GPIOs
- GMII_ENA: Must be 0
- Optionally enable internal delay (ID) mode for selected clocks using these steps:
  - DLL_RST=0: Bring DLL out of reset
  - DLL_EN=1: Start delay tuning state machine
  - DELAY_EN=1: Enable delayed clock output from DLL
  - DLL_CLK_ENA=1: Use DLL clock
- TX_CLK_CFG: Configure RGMII link speed.
  0=Disabled, 1=1000 Mbps, 2=100 Mbps, 3=10 Mbps.
  The MAC must be configured to use the same speed.
- RGMII_RX_RST=0 and RGMII_TX_RST=0: Enable RGMII interface by releasing reset
- Configure MACs to match RGMII link speed and duplex mode

4.6 RMII

This section provides information about the RMII functionality. The device includes two RMII interfaces: RMII_0 and RMII_1. RMII is a reduced MII interface with a maximum speed of 100 Mbps.

The RMII interfaces can be used by a limited number of devices:

- RMII_0: Port Module 4 or Port Module 7
- RMII_1: Port Module 8

4.6.1 RESET AND SPEED CONFIGURATION

The RMII interfaces must be enabled and assigned to a device before they can be used:

- RGMII_0_CFG and RGMII_1_CFG: Assign RMII to a device.
  These registers are shared with RGMII mode configuration
- RMII_ENA: Enable RMII ports on GPIOs
- RGMII_ENA: Must be 0
- GMII_ENA: Must be 0
- REF_CLK_SEL: Select source of 50MHz reference clock. 0=internal, 1=external.
- The CuPHY must use the REF_CLK output from GPIO6 and GPIO18 as REF_CLK in internal mode.
- RMII_RX_RST=0 and RMII_TX_RST=0: Enable RMII interface by releasing reset
- Configure MACs to match RGMII link speed and duplex mode

4.7 Port Modules

All port modules contain a Media Access Controller (MAC).

4.7.1 MEDIA ACCESS CONTROLLER (MAC)

This section provides information about the high-level functionality and the configuration options of the MAC that is used in each of the port modules.

The MAC supports the following speeds and duplex modes:

- Port Modules 0-4: 10/100/1000 Mbps full-duplex mode and 10/100 Mbps half-duplex mode.
- Port Modules 5-6: 10/100/1000/2500 Mbps full-duplex mode and 10/100 Mbps half-duplex mode.
- Port Modules 7-8: 10/100/1000 Mbps full-duplex mode and 10/100 Mbps half-duplex mode.

4.7.1.1 Reset

There are a number of resets in the port module. All of the resets can be set and cleared simultaneously. By default, all blocks are in the reset state. With reference to register CLOCK_CFG, the resets are as follows:

- MAC_RX_RST: Reset of the MAC receiver

- MAC_TX_RST: Reset of the MAC transmitter
- PORT_RST: Reset of the ingress and egress queues

When changing the MAC configuration, the port must go through a reset cycle. This is done by writing register CLOCK_CFG twice. On the first write, the reset bits are set. On the second write, the reset bits are cleared. Bits that are not reset bits in CLOCK_CFG must keep their new value for both writes.

4.7.1.2 Port Mode Configuration

The MAC provides a number of handles for configuring the port mode. With reference to the MAC_MODE_CFG, MAC_IFG_CFG, and MAC_ENA_CFG registers, the handles are as follows:

- Duplex mode (FDX_ENA). Half or full duplex.
- Enabling transmission and reception of frames (TX_ENA/RX_ENA). Clearing RX_ENA stops the reception of frames and further frames are discarded. An ongoing frame reception is interrupted. Clearing TX_ENA stops the dequeuing of frames from the egress queues, which means that frames are held back in the egress queues. An ongoing frame transmission is completed.
- Tx-to-Tx inter-frame gap (TX_IFG)

The link speed is configured using CLOCK_CFG.LINK_SPEED with the following options:

- Link speed (CLOCK_CFG.LINK_SPEED)
  1 or 2.5 Gbps (125 MHz clock), 100 Mbps (25 MHz clock), 10 Mbps (2.5 MHz clock).

4.7.1.3 Half Duplex

A number of special configuration options are available for half-duplex (HDX) mode:

- **Seed for back-off randomizer**. Field MAC_HDX_CFG.SEED seeds the randomizer used by the backoff algorithm. Use MAC_HDX_CFG.SEED_LOAD to load a new seed value.
- **Backoff after excessive collision**. Field MAC_HDX_CFG.WEXC_DIS determines whether the MAC backs off after an excessive collision has occurred. If set, backoff is disabled after excessive collisions.
- **Retransmission of frame after excessive collision**. Field MAC_HDX_CFG.RETRY_AFTER_EXC_COL_ENA determines whether the MAC retransmits frames after an excessive collision has occurred. If set, a frame is not dropped after excessive collisions, but the backoff sequence is restarted. This is a violation of IEEE 802.3, but is useful in non-dropping half-duplex flow control operation.
- **Late collision timing**. Field MAC_HDX_CFG.LATE_COL_POS adjusts the border between a collision and a late collision in steps of 1 byte. According to IEEE 802.3, section 21.3, this border is permitted to be on data byte 56 (counting frame data from 1); that is, a frame experiencing a collision on data byte 55 is always retransmitted, but it is never retransmitted when the collision is on byte 57. For each higher LATE_COL_POS value, the border is moved 1 byte higher.
- **Rx-to-Tx inter-frame gap**. The sum of MAC_IFG_CFG.RX_IFG1 and MAC_IFG_CFG.RX_IFG2 establishes the time for the Rx-to-Tx inter-frame gap. RX_IFG1 is the first part of half-duplex Rx-to-Tx inter-frame gap. Within RX_IFG1, this timing is restarted if carrier sense (CRS) has multiple high-low transitions (due to noise). RX_IFG2 is the second part of half-duplex Rx-to-Tx inter-frame gap. Within RX_IFG2, transitions on CRS are ignored.

When enabling a port for half-duplex mode, the switch core must also be enabled (SYS::FRONT_PORT_MODE.HDX-_MODE).

4.7.1.4 Frame and Type/Length Check

The MAC supports frame lengths of up to 16 kilobytes. The maximum length accepted by the MAC is configurable in MAC_MACLEN_CFG.MAX_LEN.

The MAC allows tagged frames to be 4 bytes longer and double-tagged frames to be 8 bytes longer than the specified maximum length (MAC_TAGS_CFG.VLAN_LEN_AWR_ENA). The MAC must be configured to look for VLAN tags. By default, EtherType 0x8100 identifies a VLAN tag. In addition, a custom EtherType can be configured in MAC_TAGS_CFG.TAG_ID. The MAC can be configured to look for none, one, or two tags (MAC_TAG_CFG.VLAN_AW-R_ENA, MAC_TAG_CFG.VLAN_DBL_AWR_ENA).

The type/length check (MAC_ADV_CHK_CFG.LEN_DROP_ENA) causes the MAC to discard frames with type/length errors (in-range and out-of-range errors).

4.7.1.5 Flow Control (IEEE 802.3x)

The device supports both standard full-duplex flow control (IEEE 802.3x) and priority-based flow control (IEEE 802.1Qbb). This section describes standard full-duplex flow control, and Section 4.7.1.6, Priority-Based Flow Control (IEEE 802.1Qbb) describes priority-based flow control.

In full-duplex mode, the MAC provides independent support for transmission of pause frames and reaction to incoming pause frames. This allows for asymmetric flow control configurations.

The MAC obeys received pause frames (MAC_FC_CFG.RX_FC_ENA) by pausing the egress traffic according to the timer values specified in the pause frames. In order to evaluate the pause time in the incoming pause frames, the link speed must be specified (SYS::MAC_FC_CFG.FC_LINK_SPEED).

The transmission of pause frames is triggered by assertion of a flow control condition in the ingress queues caused by a queue filling exceeding a watermark. For more information, refer to Section 4.13.7, Ingress Pause Request Generation. The MAC handles the formatting and transmission of the pause frame. The following configuration options are available:

- Transmission of pause frames (MAC_CFG_CFG.TX_FC_ENA).
- Pause timer value used in transmitted pause frames (MAC_FC_CFG.PAUSE_VAL_CFG).
- Flow control cancellation when the ingress queues de-assert the flow control condition by transmission of a pause frame with timer value 0 (MAC_FC_CFG.ZERO_PAUSE_ENA).
- Source MAC address used in transmitted pause frames (MAC_FC_MAC_HIGH_CFG, MAC_FC_MAC_LOW_CFG).

The MAC has the option to discard incoming frames when the remote link partner is not obeying the pause frames transmitted by the MAC. The MAC discards an incoming frame if a Start-of-Frame is seen after the pause frame was transmitted. It is configurable how long reaction time is given to the link partner (MAC_FC_CFG.FC_LATENCY_CFG). The benefit of this approach is that the queue system is not risking being overloaded with frames due to a non-complying link partner.

In half-duplex mode, the MAC does not react to received pause frames. If the flow control condition is asserted by the ingress queues, the industry-standard backpressure mechanism is used. Together with the ability to retransmit frames after excessive collisions (MAC_HDX_CFG.RETRY_AFTER_EXC_COL_ENA), this enables non-dropping half-duplex flow control.

4.7.1.6 Priority-Based Flow Control (IEEE 802.1Qbb)

The device supports priority-based flow control on all ports for all QoS classes.

The device provides independent support for transmission of pause frames and reaction to incoming pause frames, which allows asymmetric flow control configurations.

The device obeys received pause frames per priority (ANA::PFC_CFG.RX_PFC_ENA) by pausing the egress traffic according to the timer values specified in the pause frames. Transmission of frames belonging to QoS class n is paused if bit *n* is set in the priority_enable_vector in the incoming pause frame. The pause time for QoS class *n* is given by time[*n*] from the pause frame. The link speed must be specified in order to evaluate the pause times (ANA::PFC_CFG.FC_LINK_SPEED).

The transmission of priority-based pause frames is triggered by assertion of a flow control condition in the ingress queues caused by the memory consumption for a priority for an ingress port exceeding the BUF_Q_RSRV_I watermark. The MAC handles the formatting and transmission of the priority-based pause frame. The following configuration options are available:

- Transmission of priority-based pause frames per QoS class (QSYS::SWITCH_PORT-MODE.TX_PFC_ENA).
- Pause timer value used in transmitted priority-based pause frames (SYS::MAC_FC_CFG.PAUSE_VAL_CFG). All congested priorities use the same pause timer value. Uncongested priorities use pause timer value 0.
- Source MAC address used in transmitted priority-based pause frames (MAC_FC_MAC_HIGH_CFG, MAC_FC_MAC_LOW_CFG).
- Priority protection mode (QSYS::SWITCH_PORT_MODE.TX_PFC_MODE), which enables that when a priority congests and causes a pause frame to be sent, then the pause frame will also pause all lower priorities.

All transmitted priority-based pause frames have the priority_enable_vector set to 0xFF, independently of whether a priority is enabled for flow control. However, the pause timer value, time[*n*], is always 0 for disabled priorities.

The MAC generates and transmits a priority-based pause frames whenever a queue is congested. The device prevents excessive pause frame generation by waiting half the pause timer value between transmissions of pause frames.

When an ingress queue de-asserts the flow control condition, the MAC does not generate a priority-based pause frame with pause timer 0 for the priority. Instead, the timer in the link partner must expire. However, if another queue asserts flow control, then a priority-based pause frame is generated for that priority and for all uncongested queues a pause timer 0 is signaled to the link partner.

4.7.1.7 Frame Aging

The MAC supports frame aging where frames are discarded if a maximum transit delay through the switch is exceeded. All frames, including CPU-injected frames, are subject to aging. The transit delay is time from when a frame is fully received until that frame is scheduled for transmission through the egress MAC. The maximum allowed transit delay is configured in SYS::FRM_AGING.

Various aging modes are supported per egress port (QSYS::SWITCH_PORT_MODE.AGING_MODE):

- Disable aging.
- Discard frames exceeding the maximum transmit delay.
- Discard frames exceeding the maximum transmit delay, except frames injected by the CPU.
- Discard all frames, which corresponds to flushing the queues associated with the port.

Frames discarded due to frame aging are counted in the c_tx_aged counter.

4.7.1.8 Preemption

Preemption facilitates deterministic networking by interrupting transmission of an ordinary low priority frame to transmit an express frame, and then resuming the transmission of the preempted frame. This support of Interspersing Express traffic (IET) over a single link is specified in IEE802.3br.

The MAC Merge sub-layer supports express traffic from the Express MAC (E-MAC) combined with preemptable traffic from the Preemptable MAC (P-MAC). This is achieved by preempting any preemptable traffic that is currently being transmitted and by preventing starting the transmission of preemptable traffic. To support the merge operation, the MAC Merge sub-layer changes the encoding of the “Start of the Frame delimiters” (SFDs) originating from the P-MAC to “Start of the mPacket Delimiter” (SMD).

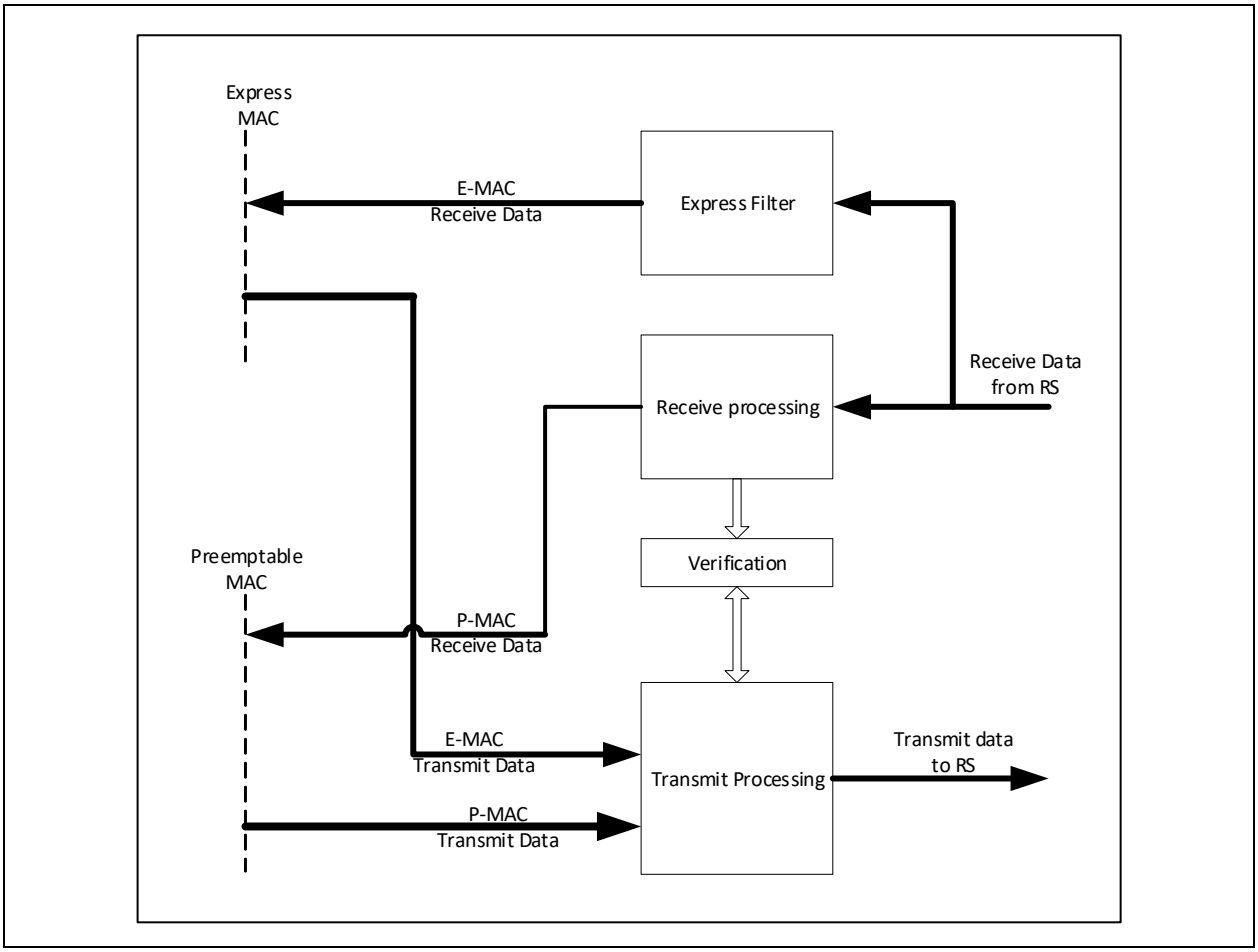
The device supports port-based preemption on all its full duplex ports.

Preemption capabilities on a port in transmit direction is enabled only if the link partner provides its support for the preemption capability through an Additional Ethernet Capabilities TLV in an LLDPDU addressed to the Nearest Bridge group address (see IEEE Std 802.1Q).

The architecture of preemption in the device is spread throughout the switch core. The necessary frame encoding and decoding are done by MAC Merge sub-layer residing in the port module.

The high level overview of the MAC Merge sub-layer can be deduced from Figure 4-2.

FIGURE 4-2: MAC MERGE OPERATION



In the receive direction, the MAC Merge Receive block decodes the SMDs of every frame and processes the frames. If the frame passes the SMD-checks enforced by receive processing block in MAC Merge sub-layer, it is delivered to the P-MAC. If the frame is received with an SFD, then it is delivered to the E-MAC. In the transmit direction, based on the indication on whether a preempted fragment or preemptable frame is being sent by P-MAC or an express frame is being sent by E-MAC, appropriate SMD is inserted in the frame and transmitted on the line. The P-MAC and E-MAC maintain their own statistics counters, see Section 4.8.1, Port Statistics for more details. Simultaneously, the MAC Merge sub-layer also maintains statistics related to assembly of the frame in Rx direction, and preemption of frame in Tx direction. Note that though both MACs maintain separate statistics, they share the same configurations.

The MAC Merge Receive and Transmit blocks are enabled by ENABLE_CONFIG.MM_RX_ENA and ENABLE_CONFIG.MM_TX_ENA after LLDP exchange has determined that the respective link partners are capable of preemption.

The previously mentioned LLPDU exchange ensures that the link partners support preemption but it does not guarantee that the underlying link supports preemption. To ensure that the link supports preemption, the verification process is performed per port. It is triggered by programming the VERIF_CONFIG register.

By programming VERIF_CONFIG::PRM_VERIFY_DIS to 0, the MAC Merge Transmit block will transmit a Verification frame and will wait for a response frame to be sent by link partner. If the response frame is received before 3 attempts of transmitting verification frame, the process is declared successful. Meanwhile, the MAC Merge Transmit block also responds to any verification frame sent by link partner, thus completing the preemption verification process of the transmit path of the link partner.

Verification process can be disabled by programming PRM_VERIFY_DIS = 1, to meet constraints on initialization time in networks where it is ensured by the design that the components are known. Only after verification process is successful or the process is disabled, would the preemption capabilities be active in the transmit direction on that port.

4.7.1.9 Redundancy Tags

The MAC supports awareness of FRER redundancy tags (R-TAGs) when RTAG48_ENA is set. When R-TAG aware, the MAC looks for EtherType = 0xF1C1 after zero or one VLAN tag. For an incoming R-TAG, the MAC deletes the 16-bit reserved field from the R-TAG. For an outgoing R-TAG, the MAC inserts a 16-bit reserved field into the R-TAG containing zeros.

When R-TAG unaware, R-TAGs are not identified and no frame modifications are performed.

4.8 Statistics

The device implements a statistics block containing the following counter groups:

- Receive statistics available per physical ingress port
- Transmit statistics available per physical egress port
- FIFO Drop statistics available per physical ingress port
- FRER statistics available per stream
- Stream filtering statistics available per stream filter
- ISDX statistics available per ISDX
- ESDX statistics available per ESDX

Each counter is 32 bits wide, which is large enough to ensure a wrap-around time longer than 30 seconds for port statistics on a 1G port. The following sub-sections list the counters.

4.8.1 PORT STATISTICS

Table 4-8 defines the per-port available Rx counters and lists the counter's addresses in the common statistics block.

TABLE 4-8: RECEIVE COUNTERS IN THE STATISTICS BLOCK

Type	Short Name	Counter Address	Description
Rx	c_rx_oct	0x00	Received octets in good and bad frames.
Rx	c_rx_uc	0x01	Number of good broadcasts.
Rx	c_rx_mc	0x02	Number of good multicasts.
Rx	c_rx_bc	0x03	Number of good unicasts.
Rx	c_rx_short	0x04	Number of short frames with valid CRC (<64 bytes).
Rx	c_rx_frag	0x05	Half-duplex links: Number of short frames with invalid CRC (<64 bytes). Full-duplex links: Number of frames received after port is paused. See SYS::FG:MAC_FC_CFG.FC_LATENCY_CFG for details. **Note:** The counter increments if pause frames are received from link partner.
Rx	c_rx_jabber	0x06	Number of long frames with invalid CRC (according to MAXLEN.MAX_LENGTH).
Rx	c_rx_crc	0x07	Number of CRC errors, alignment errors and RX_ER events.
Rx	c_rx_symbol_err	0x08	Number of frames with RX_ER events.
Rx	c_rx_sz_64	0x09	Number of 64-byte frames in good and bad frames.
Rx	c_rx_sz_65_127	0x0A	Number of 65-127-byte frames in good and bad frames.
Rx	c_rx_sz_128_255	0x0B	Number of 128-255-byte frames in good and bad frames.
Rx	c_rx_sz_256_511	0x0C	Number of 256-511-byte frames in good and bad frames.
Rx	c_rx_sz_512_1023	0x0D	Number of 512-1023-byte frames in good and bad frames.
Rx	c_rx_sz_1024_1526	0x0E	Number of 1024-1526-byte frames in good and bad frames.

TABLE 4-8: RECEIVE COUNTERS IN THE STATISTICS BLOCK (CONTINUED)

Type	Short Name	Counter Address	Description
Rx	c_rx_sz_jumbo	0x0F	Number of 1527-MAXLEN.MAX_LENGTH-byte frames in good and bad frames. Counter is only applicable if MAXLEN.MAX_LENGTH > 1526.
Rx	c_rx_pause	0x10	Number of received pause frames.
Rx	c_rx_control	0x11	Number of MAC control frames received.
Rx	c_rx_long	0x12	Number of long frames with valid CRC (according to MAXLEN.MAX-_LENGTH).
Rx	c_rx_cat_drop	0x13	Number of frames dropped due to classifier rules.
Rx	c_rx_red_prio_0	0x14	Number of received frames classified to QoS class 0 and discarded by a policer.
Rx	c_rx_red_prio_1	0x15	Number of received frames classified to QoS class 1 and discarded by a policer.
Rx	c_rx_red_prio_2	0x16	Number of received frames classified to QoS class 2 and discarded by a policer.
Rx	c_rx_red_prio_3	0x17	Number of received frames classified to QoS class 3 and discarded by a policer.
Rx	c_rx_red_prio_4	0x18	Number of received frames classified to QoS class 4 and discarded by a policer
Rx	c_rx_red_prio_5	0x19	Number of received frames classified to QoS class 5 and discarded by a policer.
Rx	c_rx_red_prio_6	0x1A	Number of received frames classified to QoS class 6 and discarded by a policer.
Rx	c_rx_red_prio_7	0x1B	Number of received frames classified to QoS class 7 and discarded by a policer.
Rx	c_rx_yellow_prio_0	0x1C	Number of received frames classified to QoS class 0 and marked yellow by a policer
Rx	c_rx_yellow_prio_1	0x1D	Number of received frames classified to QoS class 1 and marked yellow by a policer
Rx	c_rx_yellow_prio_2	0x1E	Number of received frames classified to QoS class 2 and marked yellow by a policer
Rx	c_rx_yellow_prio_3	0x1F	Number of received frames classified to QoS class 3 and marked yellow by a policer
Rx	c_rx_yellow_prio_4	0x20	Number of received frames classified to QoS class 4 and marked yellow by a policer
Rx	c_rx_yellow_prio_5	0x21	Number of received frames classified to QoS class 5 and marked yellow by a policer
Rx	c_rx_yellow_prio_6	0x22	Number of received frames classified to QoS class 6 and marked yellow by a policer
Rx	c_rx_yellow_prio_7	0x23	Number of received frames classified to QoS class 7 and marked yellow by a policer
Rx	c_rx_green_prio_0	0x24	Number of received frames classified to QoS class 0 and marked green by a policer.
Rx	c_rx_green_prio_1	0x25	Number of received frames classified to QoS class 1 and marked green by a policer.
Rx	c_rx_green_prio_2	0x26	Number of received frames classified to QoS class 2 and marked green by a policer.
Rx	c_rx_green_prio_3	0x27	Number of received frames classified to QoS class 3 and marked green by a policer.
Rx	c_rx_green_prio_4	0x28	Number of received frames classified to QoS class 4 and marked green by a policer.

TABLE 4-8: RECEIVE COUNTERS IN THE STATISTICS BLOCK (CONTINUED)

Type	Short Name	Counter Address	Description
Rx	c_rx_green_prio_5	0x29	Number of received frames classified to QoS class 5 and marked green by a policer.
Rx	c_rx_green_prio_6	0x2A	Number of received frames classified to QoS class 6 and marked green by a policer.
Rx	c_rx_green_prio_7	0x2B	Number of received frames classified to QoS class 7 and marked green by a policer.
Rx	c_rx_assembly_err	0x2C	Number of P-MAC frames with reassembly errors
Rx	c_rx_smd_err	0x2D	Number of received P-MAC frames / P-MAC frame fragments rejected due to unknown SMD value or arriving with an SMD-C when no frame is in progress
Rx	c_rx_assembly_ok	0x2E	Number of MAC frames that were successfully reassembled and delivered to P-MAC
Rx	c_rx_merge_frag	0x2F	Number of additional mPackets received due to preemption
Rx	c_rx_pmac_oct	0x30	Received octets in good and bad frames in PMAC
Rx	c_rx_pmac_uc	0x31	Number of good unicasts received by PMAC
Rx	c_rx_pmac_mc	0x32	Number of good multicasts received by PMAC
Rx	c_rx_pmac_bc	0x33	Number of good broadcasts received by PMAC
Rx	c_rx_pmac_short	0x34	Number of short frames with valid CRC (<64 bytes) received by PMAC
Rx	c_rx_pmac_frag	0x35	Number of short frames with invalid CRC (<64 bytes) received by PMAC
Rx	c_rx_pmac_jabber	0x36	Number of long frames with invalid CRC (according to MAXLEN.MAX_LENGTH) received by PMAC
Rx	c_rx_pmac_crc	0x37	Number of CRC errors, alignment errors and RX_ER events received by PMAC
Rx	c_rx_pmac_symbol-_err	0x38	Number of frames with RX_ER events received by PMAC
Rx	c_rx_pmac_sz_64	0x39	Number of 64-byte frames in good and bad frames received by PMAC
Rx	c_rx_p-mac_sz_65_127	0x3A	Number of 65-127-byte frames in good and bad frames received by PMAC
Rx	c_rx_p-mac_sz_128_255	0x3B	Number of 128-255-byte frames in good and bad frames received by PMAC
Rx	c_rx_p-mac_sz_256_511	0x3C	Number of 256-511-byte frames in good and bad frames received by PMAC
Rx	c_rx_p-mac_sz_512_1023	0x3D	Number of 512-1023-byte frames in good and bad frames received by PMAC.
Rx	c_rx_p-mac_sz_1024_1526	0x3E	Number of 1024-1526-byte frames in good and bad frames received by PMAC.
Rx	c_rx_pmac_sz_jumbo	0x3F	Number of 1527-MAXLEN.MAX_LENGTH-byte frames in good and bad frames. Counter is only applicable if MAXLEN.MAX_LENGTH > 1526
Rx	c_rx_pmac_pause	0x40	Number of received pause frames by PMAC
Rx	c_rx_pmac_control	0x41	Number of MAC control frames received by PMAC
Rx	c_rx_pmac_long	0x42	Number of long frames with valid CRC (according to MAXLEN.MAX-_LENGTH) received by PMAC

Table 4-9 defines the per-port available Tx counters and lists the counter addresses.

TABLE 4-9: TX COUNTERS IN THE STATISTICS BLOCK

Type	Short Name	Counter Address	Description
Tx	c_tx_oct	0x80	Transmitted octets in good and bad frames.
Tx	c_tx_uc	0x81	Number of good unicasts.
Tx	c_tx_mc	0x82	Number of good multicasts.
Tx	c_tx_bc	0x83	Number of good broadcasts.
Tx	c_tx_col	0x84	Number of transmitted frames experiencing a collision. An excessive collided frame gives 16 counts.
Tx	c_txdrop	0x85	Number of frames dropped due to excessive collisions or late collisions.
Tx	c_txpause	0x86	Number of transmitted pause frames.
Tx	c_tx_sz_64	0x87	Number of 64-byte frames in good and bad frames.
Tx	c_tx_sz_65_127	0x88	Number of 65-127-byte frames in good and bad frames.
Tx	c_tx_sz_128_255	0x89	Number of 128-255-byte frames in good and bad frames.
Tx	c_tx_sz_256_511	0x8A	Number of 256-511-byte frames in good and bad frames.
Tx	c_tx_sz_512_1023	0x8B	Number of 512-1023-byte frames in good and bad frames.
Tx	c_tx_sz_1024_1526	0x8C	Number of 1024-1526-byte frames in good and bad frames.
Tx	c_tx_sz_jumbo	0x8D	Number of 1527-MAXLEN.MAX_LENGTH-byte frames in good and bad frames.
Tx	c_tx_yellow_prio_0	0x8E	Number of transmitted frames classified to QoS class 0 with DP level 1.
Tx	c_tx_yellow_prio_1	0x8F	Number of transmitted frames classified to QoS class 1 with DP level 1.
Tx	c_tx_yellow_prio_2	0x90	Number of transmitted frames classified to QoS class 2 with DP level 1.
Tx	c_tx_yellow_prio_3	0x91	Number of transmitted frames classified to QoS class 3 with DP level 1.
Tx	c_tx_yellow_prio_4	0x92	Number of transmitted frames classified to QoS class 4 with DP level 1.
Tx	c_tx_yellow_prio_5	0x93	Number of transmitted frames classified to QoS class 5 with DP level 1.
Tx	c_tx_yellow_prio_6	0x94	Number of transmitted frames classified to QoS class 6 with DP level 1.
Tx	c_tx_yellow_prio_7	0x95	Number of transmitted frames classified to QoS class 7 with DP level 1.
Tx	c_tx_green_prio_0	0x96	Number of transmitted frames classified to QoS class 0 with DP level 0.
Tx	c_tx_green_prio_1	0x97	Number of transmitted frames classified to QoS class 1 with DP level 0.
Tx	c_tx_green_prio_2	0x98	Number of transmitted frames classified to QoS class 2 with DP level 0.
Tx	c_tx_green_prio_3	0x99	Number of transmitted frames classified to QoS class 3 with DP level 0.
Tx	c_tx_green_prio_4	0x9A	Number of transmitted frames classified to QoS class 4 with DP level 0.
Tx	c_tx_green_prio_5	0x9B	Number of transmitted frames classified to QoS class 5 with DP level 0.
Tx	c_tx_green_prio_6	0x9C	Number of transmitted frames classified to QoS class 6 with DP level 0.
Tx	c_tx_green_prio_7	0x9D	Number of transmitted frames classified to QoS class 7 with DP level 0.
Tx	c_tx_aged	0x9E	Number of frames dropped due to frame aging.
Tx	c_tx_llct	0x9F	RESERVED
Tx	c_tx_ct	0xA0	Number of frames cut-through forwarded.
Tx	c_tx_buf_drop	0xA1	Number of frames discarded by queue system
Tx	c_tx_mm_hold	0xA2	Number of times MM_CTL.request(HOLD) primitive was received by PMAC
Tx	c_tx_merge_frag	0xA3	Number of additional mPackets transmitted by PMAC due to preemption
Tx	c_tx_pmac_oct	0xA4	Transmitted octets in good and bad frames by PMAC
Tx	c_tx_pmac_uc	0xA5	Number of good unicasts by PMAC
Tx	c_tx_pmac_mc	0xA6	Number of good multicasts by PMAC

TABLE 4-9: TX COUNTERS IN THE STATISTICS BLOCK (CONTINUED)

Type	Short Name	Counter Address	Description
Tx	c_tx_pmac_bc	0xA7	Number of good broadcasts by PMAC
Tx	c_tx_pmac_pause	0xA8	Number of transmitted pause frames by PMAC
Tx	c_tx_pmac_sz_64	0xA9	Number of 64-byte frames in good and bad frames transmitted by PMAC
Tx	c_tx_p-mac_sz_65_127	0xAA	Number of 65-127-byte frames in good and bad frames transmitted by PMAC
Tx	c_tx_p-mac_sz_128_255	0xAB	Number of 128-255-byte frames in good and bad frames transmitted by PMAC
Tx	c_tx_p-mac_sz_256_511	0xAC	Number of 256-511-byte frames in good and bad frames transmitted by PMAC
Tx	c_tx_p-mac_sz_512_1023	0xAD	Number of 512-1023-byte frames in good and bad frames transmitted by PMAC
Tx	c_tx_p-mac_sz_1024_1526	0xAE	Number of 1024-1526-byte frames in good and bad frames transmitted by PMAC
Tx	c_tx_pmac_sz_jumbo	0xAF	Number of 1527-MAXLEN.MAX_LENGTH-byte frames in good and bad frames transmitted by PMAC

Table 4-10 defines the per-port available FIFO drop counters and lists the counter addresses.

TABLE 4-10: FIFO DROP COUNTERS IN THE STATISTICS BLOCK

Type	Short Name	Counter Address	Description
Drop	c_dr_local	0x100	Number of frames discarded due to no destinations.
Drop	c_dr_tail	0x101	Number of frames discarded due to no more memory in the queue system (tail drop).
Drop	c_dr_yellow_prio_0	0x102	Number of FIFO discarded frames classified to QoS class 0 with DP level 1
Drop	c_dr_yellow_prio_1	0x103	Number of FIFO discarded frames classified to QoS class 1 with DP level 1
Drop	c_dr_yellow_prio_2	0x104	Number of FIFO discarded frames classified to QoS class 2 with DP level 1
Drop	c_dr_yellow_prio_3	0x105	Number of FIFO discarded frames classified to QoS class 3 with DP level 1
Drop	c_dr_yellow_prio_4	0x106	Number of FIFO discarded frames classified to QoS class 4 with DP level 1
Drop	c_dr_yellow_prio_5	0x107	Number of FIFO discarded frames classified to QoS class 5 with DP level 1
Drop	c_dr_yellow_prio_6	0x108	Number of FIFO discarded frames classified to QoS class 6 with DP level 1
Drop	c_dr_yellow_prio_7	0x109	Number of FIFO discarded frames classified to QoS class 7 with DP level 1
Drop	c_dr_green_prio_0	0x10A	Number of FIFO discarded frames classified to QoS class 0 with DP level 0.
Drop	c_dr_green_prio_1	0x10B	Number of FIFO discarded frames classified to QoS class 1 with DP level 0.
Drop	c_dr_green_prio_2	0x10C	Number of FIFO discarded frames classified to QoS class 2 with DP level 0.
Drop	c_dr_green_prio_3	0x10D	Number of FIFO discarded frames classified to QoS class 3 with DP level 0.

TABLE 4-10: FIFO DROP COUNTERS IN THE STATISTICS BLOCK (CONTINUED)

Type	Short Name	Counter Address	Description
Drop	c_dr_green_prio_4	0x10E	Number of FIFO discarded frames classified to QoS class 4 with DP level 0.
Drop	c_dr_green_prio_5	0x10F	Number of FIFO discarded frames classified to QoS class 5 with DP level 0.
Drop	c_dr_green_prio_6	0x110	Number of FIFO discarded frames classified to QoS class 6 with DP level 0.
Drop	c_dr_green_prio_7	0x111	Number of FIFO discarded frames classified to QoS class 7 with DP level 0.

4.8.2 FRER SEQUENCE GENERATOR STATISTICS

Table 4-11 defines the per-ISDX available FRER sequence generator counters in the statistics block and lists the counter addresses.

TABLE 4-11: FRER STREAM COUNTERS IN THE STATISTICS BLOCK

Type	Short Name	Counter Address	Description
FRER	c_st_sid_in_pkt	0x180	Incremented once per packet identified by the ISDX.

4.8.3 STREAM FILTER STATISTICS

Table 4-12 defines the stream filter counters available per Stream Filter ID (SFID) in the statistics block and lists the counter addresses.

TABLE 4-12: STREAM FILTER COUNTERS IN THE STATISTICS BLOCK

Type	Short Name	Counter Address	Description
Stream Filter	c_sf_matching_-frames_count	0x200	Number of frames matching the filter and its specifications.
Stream Filter	c_sf_not_passing_-frames_count	0x201	Number of frames those have not passed the stream gate associated with filter.
Stream Filter	c_sf_not_passing_s-du_count	0x202	Number of frames those have not passed the MAX SDU check associated with filter.
Stream Filter	c_sf_red_frames_count	0x203	Number of frames those have been dropped by policer pointed to by filter.
Stream Filter	c_sf_stream_-block_count	0x204	Number of frames blocked by stream filter.

Note: The MAX SDU Filter check is not performed on cut through frames, so c_sf_not_passing_sdu_count is always zero for those frames.

4.8.4 ISDX AND ESDX STATISTICS

ISDX and ESDX statistics work on frames per color. The color is given as the DP level after the frames are policed by the PSFP policer, port policer and Versatile Content-Aware Processor (VCAP) policer (see Section 4.10, "VCAP" for more details):

- Green frames: Frames with DP level 0 after the combined policer action.
- Yellow frames: Frames with DP level 1 after the combined policer action.
- Red frames: Frames discarded by the combined policer action.

Table 4-13 defines the ISDX counters available per ISDX in the statistics block and lists the counter addresses.

TABLE 4-13: ISDX COUNTERS IN THE STATISTICS BLOCK

Type	Short Name	Counter Address	Description
ISDX	c_is_green_oct	0x280	Number of octets in green frames classified to ISDX that are forwarded by the device to one or more destination ports.
ISDX	c_is_green_pkt	0x281	Number of green frames classified to ISDX that are forwarded by the device to one or more destination ports.
ISDX	c_is_yellow_oct	0x282	Number of octets in yellow frames classified to ISDX that are forwarded by the device to one or more destination ports.
ISDX	c_is_yellow_pkt	0x283	Number of yellow frames classified to ISDX that are forwarded by the device to one or more destination ports.
ISDX	c_is_red_oct	0x284	Number of octets in received frames classified to ISDX that are discarded by a policer.
ISDX	c_is_red_pkt	0x285	Number of received frames classified to ISDX that are discarded by a policer.
ISDX	c_is_drop_green_oct	0x286	Number of octets in green frames classified to ISDX that are not discarded by a policer but discarded by the queue system or the analyzer.
ISDX	c_is_drop_green_pkt	0x287	Number of green frames classified to ISDX that are not discarded by a policer but discarded by the queue system or the analyzer.
ISDX	c_is_drop_yellow_oct	0x288	Number of octets in yellow frames classified to ISDX that are not discarded by a policer but discarded by the queue system or the analyzer.
ISDX	c_is_drop_yellow_pkt	0x289	Number of yellow frames classified to ISDX that are not discarded by a policer but discarded by the queue system or the analyzer.

Only one set of the ISDX counters (frames and octets) is triggered per frame. For instance, if the c_is_green_pkt and c_is_green_oct counters are incrementing, it implies that the frame is forwarded by the device and that it is not discarded by any of the policers, the queue system, or the analyzer.

The ISDX drop counters are triggered if a frame is discarded by the device after the policers. The following blocks can discard a frame:

- The queue system due to congestion.
- The analyzer due to the forwarding decision (destination port set is empty).

Table 4-14 defines the per-ESDX available egress ESDX counters and lists the counter addresses.

TABLE 4-14: ESDX COUNTERS IN THE STATISTICS BLOCK

Type	Short Name	Counter Address	Description
ESDX	c_es_green_oct	0x300	Number of octets in transmitted frames classified to ESDX with DP level 0.
ESDX	c_es_green_pkt	0x301	Number of transmitted frames classified to ESDX with DP level 0.
ESDX	c_es_yellow_oct	0x302	Number of octets in transmitted frames classified to ESDX with DP level 1.
ESDX	c_es_yellow_pkt	0x303	Number of transmitted frames classified to ESDX with DP level 1.

By default, the egress ESDX counters are indexed using the ESDX from the ES0 action and in case, there is no ES0 hit, the frame's ISDX is used. This behavior can be changed (REW::STAT_CFG) with the following options on how to index the egress ESDX counters:

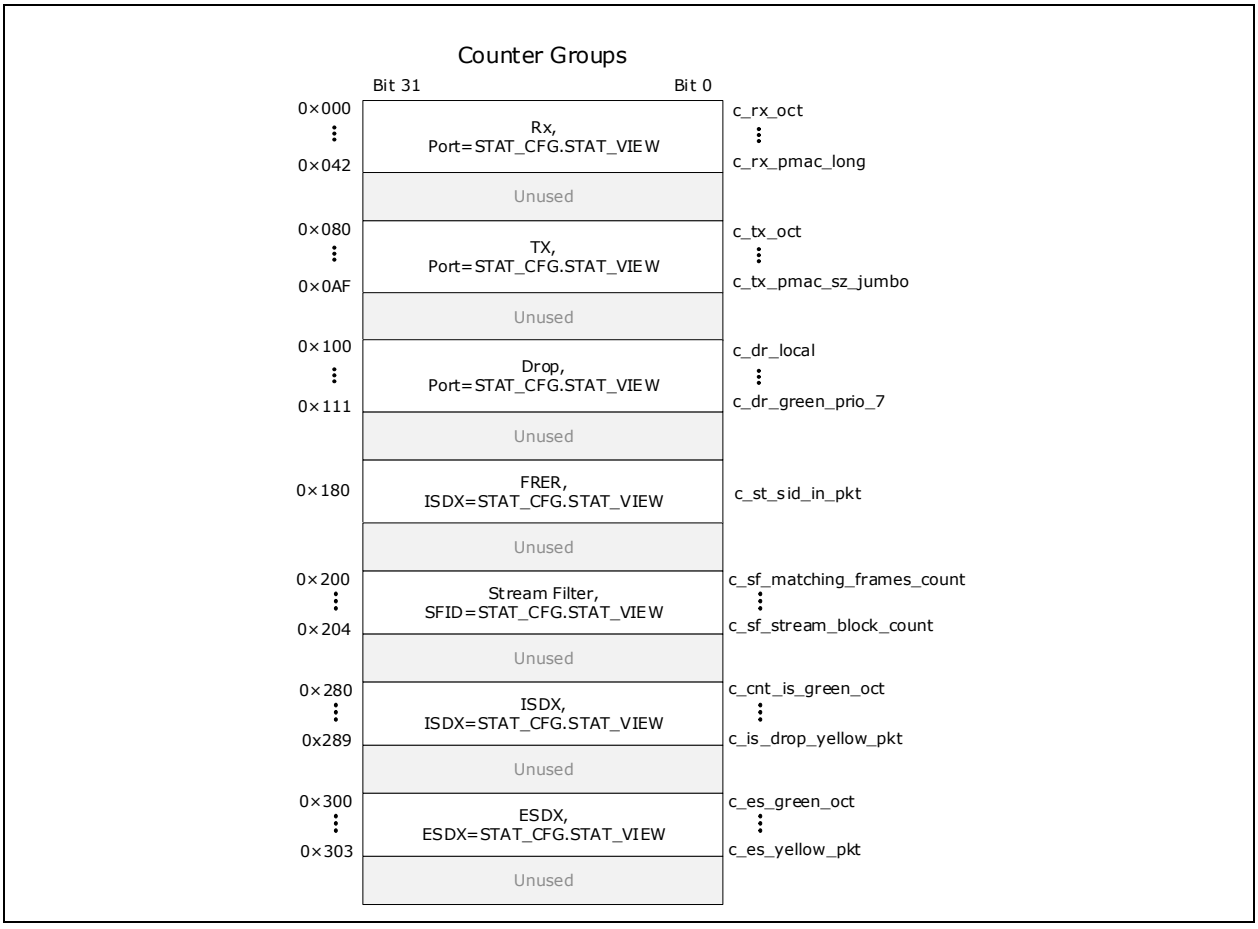
- Use the ESDX from the ES0 action. If there is no ES0 hit, use the frame's ISDX. This is the default behavior.
- Use the ESDX from the ES0 action. If there is no ES0 hit, do not count the frame in the egress ESDX statistics.
- Always use the frame's ISDX even when there is an ES0 hit. This enables per-VLAN counters when the ISDX ingress classification at the same time assigns a unique ISDX per VID.
- Use eight times the frame's source port plus the frame's QoS class. This enables per-ingress port QoS counters.

4.8.5 ACCESSING AND CLEARING COUNTERS

The counters are accessed through SYS:STAT:CNT using the counter address given for each counter in the tables in the preceding sections. Only a subset of the counters for all ports and streams are addressable at the same time. The current subset is programmed in STAT_CFG.STAT_VIEW and includes port counters (Rx, Tx, drop) and stream counters for the number programmed in STAT_CFG.STAT_VIEW.

The following illustration shows the layout of the counter subset that are addressable through SYS:STAT:CNT. To change the view, write a new number to STAT_CFG.STAT_VIEW.

FIGURE 4-3: COUNTER LAYOUT (SYS:STAT:CNT) PER VIEW



Read Example:

To read the number of good unicast frames transmitted on port 3 (counter c_tx_uc), set the current view in STAT_CFG.STAT_VIEW to 3 and read SYS:STAT:CNT[0x41]. Note that with the current view, one could for instance also read Rx and drop counters on port 3, as well as stream filter counters for SFID 3.

The counters can be cleared per counter group per view. Writing to register STAT_CFG.STAT_CLEAR_SHOT clears all counters associated with the counter groups specified by STAT_CFG.STAT_CLEAR_SHOT for the view specified in STAT_CFG.STAT_VIEW.

It is possible to select whether to count frames or bytes for the following specific counters.

- The Rx priority counters (c_rx_red_prio_*, c_rx_yellow_prio_*, c_rx_green_prio_*, where x is 0 through 7).
- The Tx priority counters (c_tx_yellow_prio_*, c_tx_green_prio_*, where x is 0 through 7).
- The Drop priority counters (c_dr_yellow_prio_*, c_dr_green_prio_*, where x is 0 through 7).

The Rx priority counters are programmed through ANA::AGENCTRL, and the Tx and drop priority counters are programmed through QSYS::STAT_CNT_CFG. When counting bytes, the frame length excluding inter frame gap and preamble is counted.

For testing purposes, all counters are both readable and writable. All counters wrap around to 0 when reaching the maximum.

4.9 Basic Classifier

The switch core includes a common basic classifier, which determines a number of properties affecting the forwarding of each frame through the switch. These properties are:

- Frame acceptance filtering. Drop illegal frame types.
- QoS classification. Assign one of eight QoS classes to the frame.
- Drop precedence (DP) classification. Assign one of two drop precedence levels to the frame.
- DSCP classification. Assign one of 64 DSCP values to the frame.
- VLAN classification. Extract tag information from the frame or use the port VLAN.
- Link aggregation code generation. Generate the link aggregation code.
- CPU forwarding determination. Determine CPU Forwarding and CPU extraction queue number

The outcome of the classifier is the basic classification result, which can be overruled by more intelligent frame processing in the VCAP IS1. For more information, see Section 4.10, VCAP.

4.9.1 GENERAL DATA EXTRACTION SETUP

This section provides information about the overall settings for data extraction that provides the data input to the classifier, VCAP, analyzer, and rewriter.

It is programmable which VLAN tags are recognized is programmable. The use of Layer-3 and Layer-4 information for classification and forwarding can also be controlled.

The device recognizes three different VLAN tags:

- Customer tags (C-TAGs), which use TPID 0x8100.
- Service tags (S-TAGs), which use TPID 0x88A8 (IEEE 802.1ad).
- Service tags (S-TAGs), which use a custom TPID programmed in ANA:COMMON:VLAN_ETYPE_CFG.

The device can parse and use information from up to two VLAN tags of any of the kinds described above.

By default, the outer VLAN tag is extracted and used for both the basic classification and the VCAP IS1 key generation. However, for both the basic classification and the VCAP IS1, there is an option to use the inner VLAN tag instead for frames with at least two VLAN tags. For basic classification, this is controlled in VLAN_CFG.VLAN_INNER_TAG_ENA and affects both QoS, DP, and VLAN classification as well as the frame acceptance filter. For IS1, this is controlled per lookup in S1_VLAN_INNER_TAG_ENA. Note that several keys in IS1 always contain both the inner VLAN tag and the outer VLAN tag.

It is also possible to configure the device to parse FRER redundancy tags (R-TAGs). This can be enabled by setting REDTAG_PARSE_CFG register per port. When the frame's Ethertype matches 0xF1C1, the packet is considered Sequenced (SEQ), otherwise the packet is considered Non-sequenced (NON_SEQ). This parsed information on whether the packet is SEQ / NON_SEQ along with the sequence number extracted from the frame is passed to analyzer, which implements the Stream Table.

An R-TAG must always be placed after any VLAN tags. The device supports up to one VLAN tag before the R-TAG. In case a VLAN tag is present in the packet after an R-TAG, then the VLAN tag is not extracted.

Various blocks in the device uses Layer-3 and Layer-4 information for classification and forwarding. Layer-3 and Layer-4 information can be extracted from a frame with up to two tags (VLAN or redundancy). Frames with more than two tags are considered non-IP frames.

The actual use of Layer-3 and Layer-4 information for classification, forwarding, and rewriting is enabled in L3_PARSE_CFG. The following blocks are affected by this functionality:

- Basic classification: QoS, DP, and DSCP classification, link aggregation code generation, CPU forwarding
- VCAP: TCAM keys (IS1, IS2) using Layer-3 and Layer-4 information
- Analyzer: Flooding and forwarding of IP multicast frames
- Rewriter: Rewriting of IP information

4.9.2 FRAME ACCEPTANCE FILTERING

Based on the configurations in the DROP_CFG and PORT_MISC registers, the classifier instructs the queue system to drop or forward certain frames types, such as:

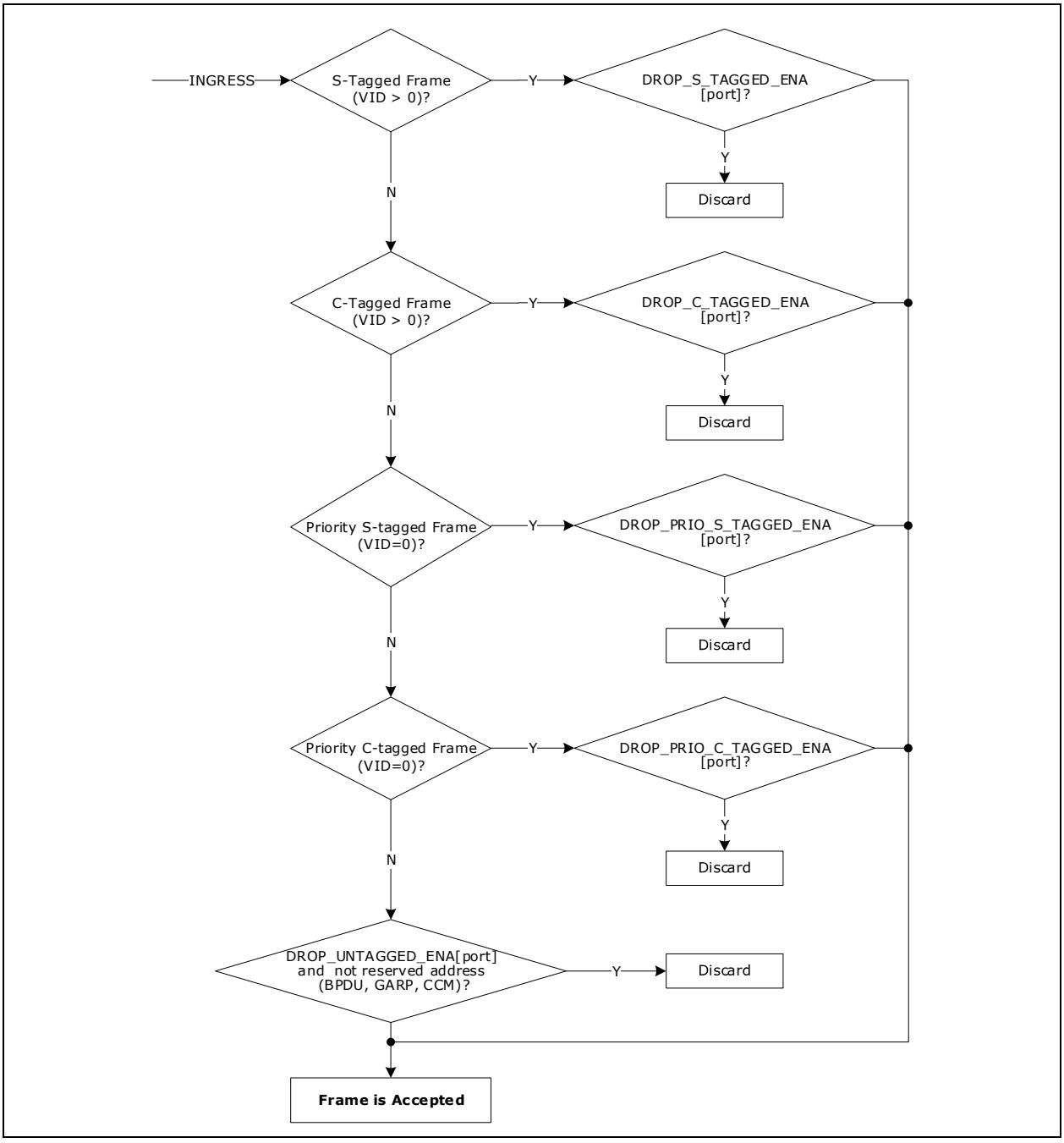
- Frames with a multicast source MAC address
- Frames with a null source or null destination MAC address (address = 0x000000000000)
- Frames with errors signaled by the MAC (for example, an FCS error)
- MAC control frames
- Pause frames after flow control processing in the MAC.
- Untagged frames (excluding frames with reserved destination MAC addresses from the BPDU, GARP, and Link trace/CCM address ranges).
- Priority S-tagged frames
- Priority C-tagged frames
- VLAN S-tagged frames
- VLAN C-tagged frames

By default, MAC control frames, pause frames, and frames with errors are dropped by the classifier.

The VLAN acceptance filter decides whether a frame's VLAN tagging is allowed on the port. By default, the outer VLAN tag is used as input to the filter, however, there is an option to use the inner VLAN tag instead for double tagged frames.

Figure 4-4 shows the flowchart for the VLAN acceptance filter.

FIGURE 4-4: VLAN ACCEPTANCE FILTER



If the frame is accepted by the VLAN acceptance filter, it can still be discarded in other places of the switch, such as the following:

- Policers, due to traffic exceeding a peak information rate.
- IS2 Security TCAM, due to permit/deny rules.
- Analyzer, due to forwarding decisions such as VLAN ingress filtering.
- Queue system, due to lack of resources, frame aging, or excessive collisions.

4.9.3 QOS, DP, AND DSCP CLASSIFICATION

This section provides information about the functions in the QoS, DP, and DSCP classification. The three tasks are described as one, because the tasks have a significant amount of functionality in common.

The basic classification provides the user with control of the QoS, DP, and DSCP classification algorithm. The result of the basic classification are the following frame properties, which follow the frame through the switch:

- The frame's QoS class. This class is encoded in a 3-bit field, where 7 is the highest priority QoS class and 0 is the lowest priority QoS class. The QoS class is used by the queue system when enqueuing frames and when evaluating resource consumptions, for policing, statistics, and rewriter actions.
- The frame's internal priority value (IPV). By default, the IPV is set to the same value as the frame's QoS class. The IPV may be changed by a stream gate. For more details, see Section 4.11.7, Stream Gates.
- The frame's DP level. This level is encoded in a 1-bit field, where frames with DP = 1 have the highest probability of being dropped and frames with DP = 0 have the lowest probability. The DP level is used by the dual leaky bucket policers for measuring committed and peak information rates, for restricting memory consumptions in the queue system, for collecting statistics, and for rewriting priority information in the rewriter. The DP level is incremented by the policers if a frame is exceeding a programmed committed information rate.
- The frame's DSCP. This value is encoded in a 6-bit fields. The DSCP value is forwarded with the frame to the rewriter where it is translated and rewritten into the frame. The DSCP value is only applicable to IPv4 and IPv6 frames.

The classifier looks for the following fields in the incoming frame to determine the QoS, DP, and DSCP classification:

- Port default QoS class and DP level. The default DSCP value is the frame's DSCP value. For non-IP frames, the DSCP is 0 and it is not used elsewhere in the switch.
- Priority Code Point (PCP) when the frame is VLAN tagged or priority tagged. There is an option to use the inner tag for double tagged frames (VLAN_CFG.VLAN_INNER_TAG_ENA). Both S-tagged and C-tagged frames are considered.
- Drop Eligible Indicator (DEI) when the frame is VLAN tagged or priority tagged. There is an option to use the inner tag for double tagged frames (VLAN_CFG.VLAN_INNER_TAG_ENA). Both S-tagged and C-tagged frames are considered.
- DSCP (all 6 bits, both for IPv4 and IPv6 packets). The classifier can look for the DSCP value behind up to two VLAN tags.

Figure 4-5 shows the flow chart of basic QoS and DP classification.

FIGURE 4-5: QOS AND DP BASIC CLASSIFICATION FLOW

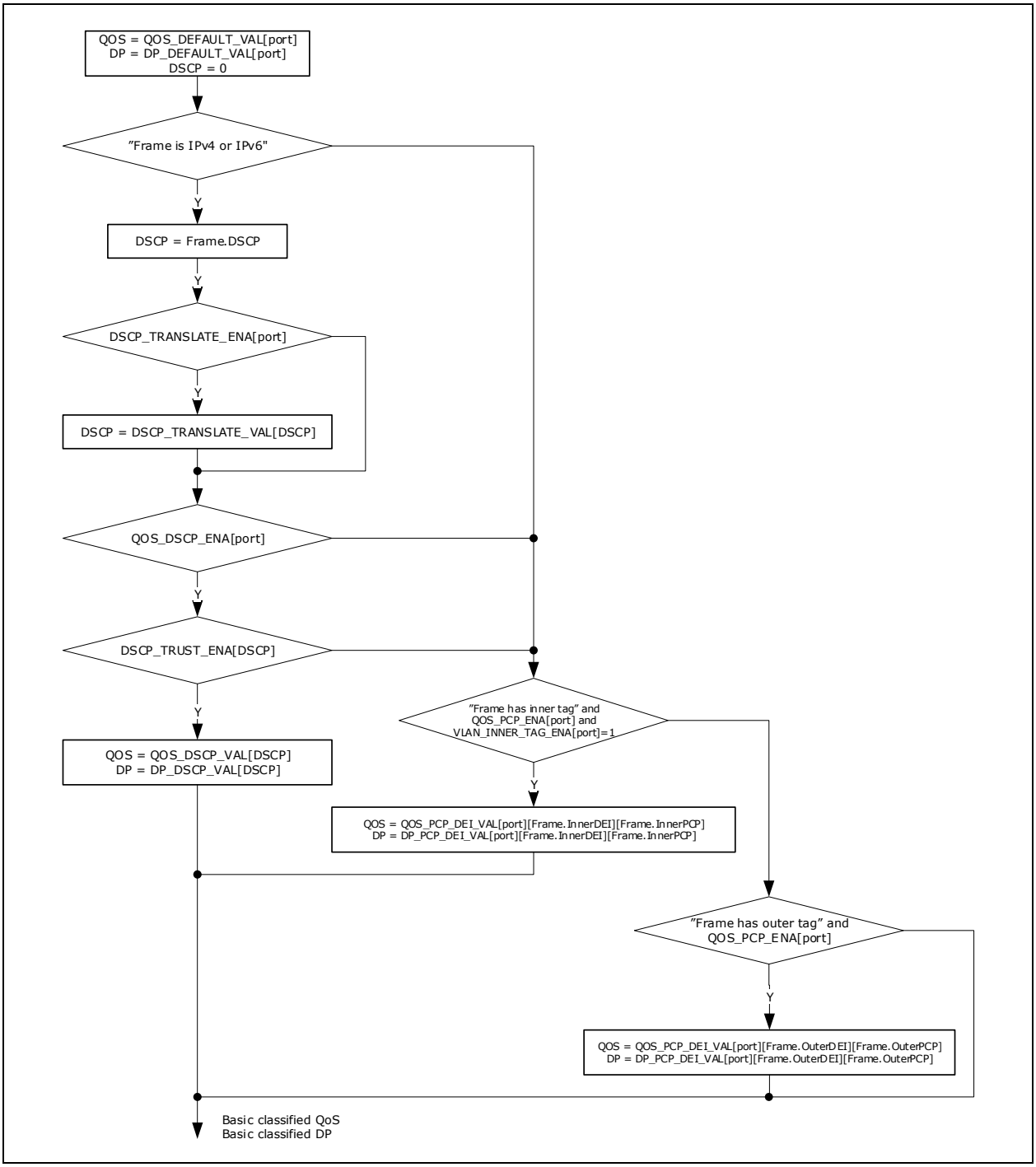
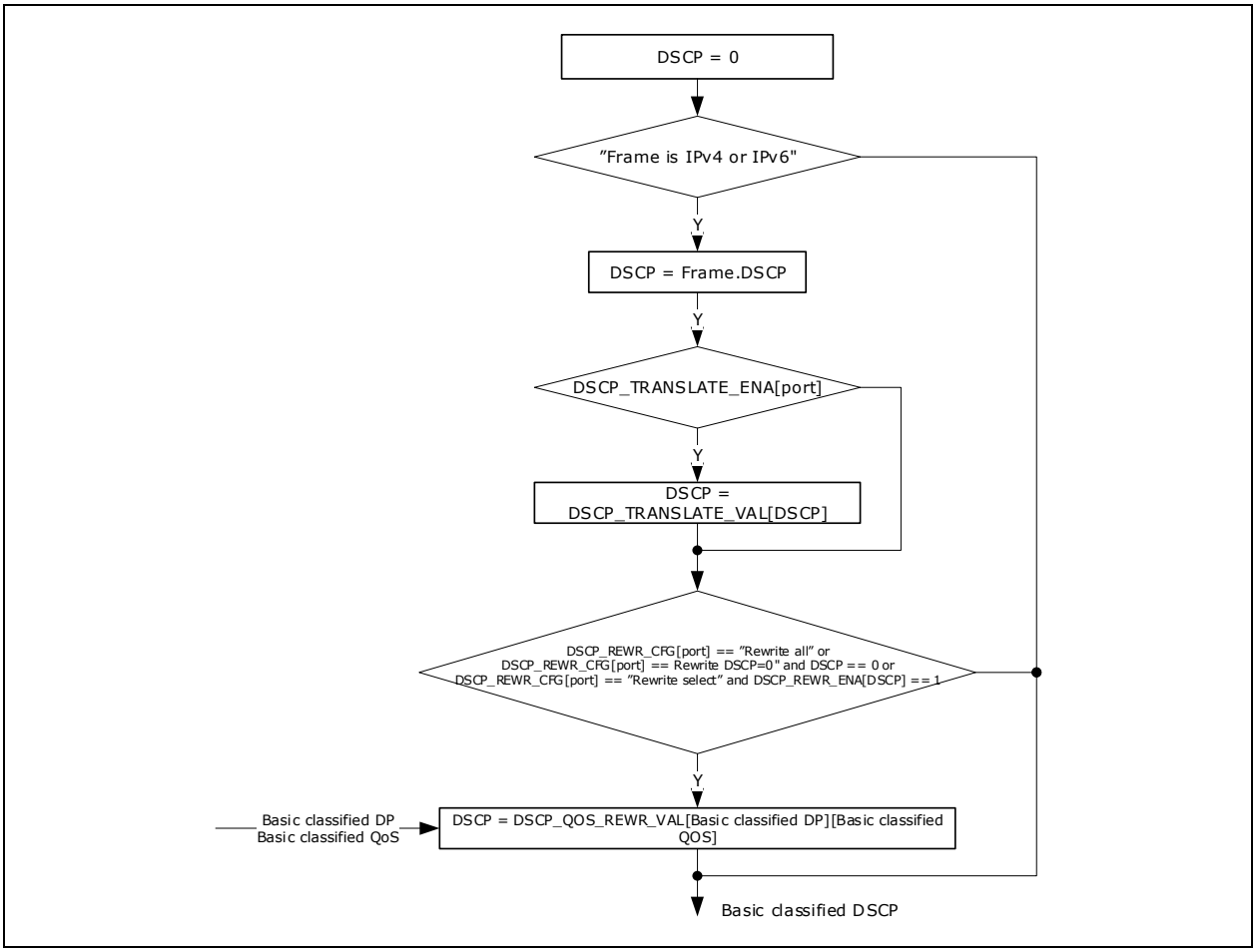


Figure 4-6 shows the flow chart for basic DSCP classification.

FIGURE 4-6: BASIC DSCP CLASSIFICATION FLOW CHART



The translation part of the DSCP classification is common for both QoS, DP, and DSCP classification.

The basic classified QoS, DP, and DSCP can be overwritten by more intelligent decisions made in the VCAP IS1.

4.9.4 VLAN CLASSIFICATION

The VLAN classification determines a tag header for all frames. The tag header includes the following information:

- Priority Code Point (PCP)
- Drop Eligible Indicator (DEI)
- VLAN Identifier (VID)
- Tag Protocol Identifier (TPID) type (TAG_TYPE). This field informs whether tag used for classification was a C-tag or an S-tag.

The tag header determined by the classifier is carried with the frame through the switch and is used in various places such as the analyzer for forwarding and the rewriter for egress tagging operations.

The device recognizes three kinds of tags based on the TPID, which is the EtherType in front of the tag:

- Customer tags (C-TAGs), which use TPID 0x8100.
- Service tags (S-TAGs), which use TPID 0x88A8 (IEEE 802.1ad).
- Service tags (S-TAGs), which use a custom TPID programmed in SYS::VLAN_ETYPE_CFG.

For customer tags and service tags, both VLAN tags (tags with nonzero VID) and priority tags (tags with VID = 0) are processed.

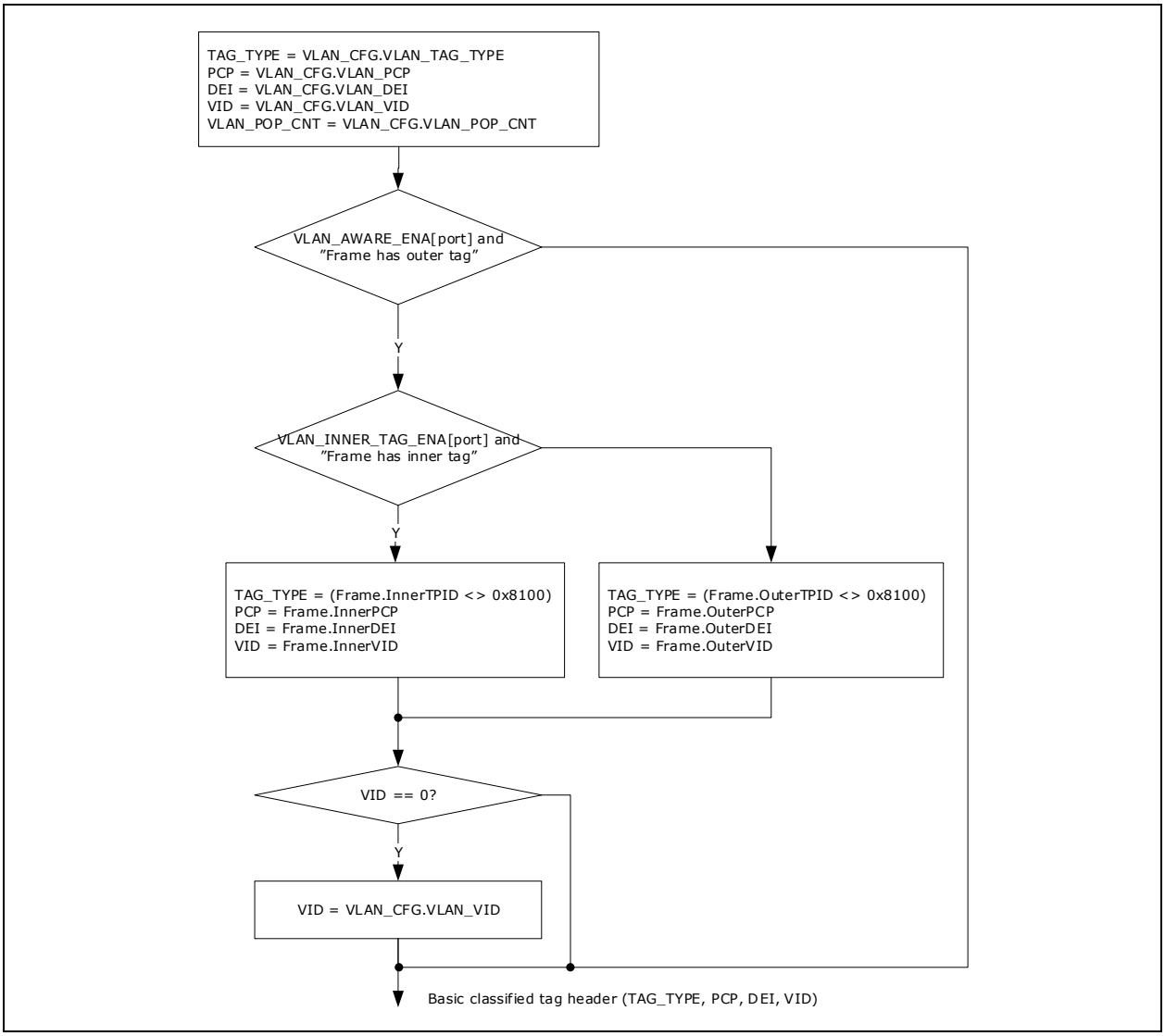
The tag header is either retrieved from a tag in the incoming frame or from a default port-based tag header. The port-based tag header is configured in ANA:PORT:VLAN_CFG.

For double tagged frames, there is an option to use the inner tag instead of the outer tag (VLAN_CFG.VLAN_INNER_TAG_ENA).

In addition to the tag header, the port decides the number of VLAN tags to pop at egress (VLAN_POP_CNT). If the configured number of tags to pop is greater than the actual number of tags in the frame, the number is reduced to the number of actual tags in the frame.

Figure 4-7 shows the flow chart for basic VLAN classification.

FIGURE 4-7: BASIC VLAN CLASSIFICATION FLOW



The basic classified tag header can be overwritten by more intelligent decisions made in the VCAP IS1.

4.9.5 LINK AGGREGATION CODE GENERATION

This section provides information about the functions in link aggregation code generation.

The classifier generates a link aggregation code, which is used in the analyzer when selecting to which port in a link aggregation group a frame is forwarded.

The following contributions to the link aggregation code is configured in the AGGR_CFG register:

- Destination MAC address—use the lower 12 bits of the DMAC.

- Source MAC address—use the lower 12 bits of the SMAC.
- IPv6 flow label—use the 20 bits of the flow label.
- IPv4 source and destination IP addresses—use the lower 8 bits of the SIP and DIP.
- TCP/UDP source and destination port for IPv4 and IPv6 frames—use the lower 8 bits of the SPORT and DPORT.
- Random aggregation code—use a pseudo-random number instead of the frame information.

Each of the enabled contributions are XOR'ed together, yielding a 4-bit aggregation code ranging from 0 to 15.

4.9.6 CPU FORWARDING DETERMINATION

The classifier has support for determining whether certain frames must be forwarded to the CPU extraction queues. Other parts of the device can also determine CPU forwarding, for example the analyzer or the VCAP IS2. All events leading to CPU forwarding are OR'ed together, and the final CPU extraction queue mask, which is available to the user, contains the sum of all events leading to CPU extraction.

Upon CPU forwarding by the classifier, the frame type determines whether the frame is redirected or copied to the CPU. Any frame type or event causing a redirection to the CPU cause all front ports to be removed from the forwarding decision - only the CPU receives the frame. When copying a frame to the CPU, the normal forwarding of the frame is unaffected.

Table 4-15 lists the standard frame types, with respect to CPU forwarding, that are recognized by the classifier.

TABLE 4-15: FRAME TYPE DEFINITIONS FOR CPU FORWARDING

Frame	Condition	Copy/Redirect
BPDU frames. Reserved Addresses (IEEE 802.1D 7.12.6)	DMAC = 0x0180C2000000 to 0x0180C20000F (BPDUs and various Slow protocols supporting spanning tree, link aggregation, port authentication)	Redirect/Copy/Discard
Reserved ALLBRIDGE address	DMAC = 0x0180C2000010	Redirect/Copy/Discard
GARP Application Addresses (IEEE 802.1D 12.5)	DMAC = 0x0180C2000020 to 0x0180C200002F	Redirect/Copy/Discard
CCM/Link Trace Addresses (IEEE P802.1ag)	DMAC = 0x0180C2000030 to 0x0180C200003F	Redirect/Copy/Discard
IGMP	DMAC = 0x01005E000000 to 0x01005E7FFFFF EtherType = IPv4 IP Protocol = IGMP	Redirect
MLD	DMAC = 0x333300000000 to 0x3333FFFFFFFF EtherType = IPv6 IPv6 Next Header = 0 Hop-by-hop options header with the first option being a Router Alert option with the MLD message (Option Type = 5, Opt Data Len = 2, Option Data = 0).	Redirect
IPv4 Multicast Ctrl	DMAC = 0x01005E000000 to 0x01005E7FFFFF EtherType = IPv4 IP protocol is not IGMP IPv4 DIP inside 224.0.0.x	Copy
Source port	All frames received on enabled ingress port	Copy
All other frames		

In addition, the classifier can recognize Versatile Register Access Protocol (VRAP) frames and redirect such frames to the CPU. This is a proprietary frame format, which is used for reading and writing switch configuration registers through Ethernet frames, see Section 4.19, VRAP Engine.

The VRAP filter in the classifier performs three checks in order to determine whether a frame is a VRAP frame:

1. VLAN check. The filter can be either VLAN unaware or VLAN aware (ANA::VRAP_CFG.VRAP_VLAN_AWARE_ENA). If VLAN unaware, VRAP frames must be untagged. If VLAN aware, VRAP frames must be VLAN tagged and the frame's VID must match a configured value (ANA::VRAP_CFG.VRAP_VID). Double VLAN tagged frames always fail this check.
2. EtherType and EPID check. The EtherType must be 0x8880 and the EPID must be 0x0004 (bytes 0 and 1 after

the EtherType).

3. VRAP header check. The VRAP header (bytes 0, 1, 2, and 3 after the EPID) must match a 32-bit configured value (ANA::VRAP_HDR_DATA) where any bits can be don't cared by a mask (ANA::VRAP_HDR_MASK).

If all three checks are fulfilled, frames are redirected to CPU extraction queue ANA::CPUQ_CFG2.CPUQ_VRAP.

The VRAP filter is enabled in ANA:PORT:CPU_FWD_CFG.

4.10 VCAP

The Versatile Content-Aware Processor (VCAP) is a hardware-accelerated packet classification engine designed for wire-speed frame inspection and processing. It enables advanced functionality such as VLAN and QoS classification, IP source guard, and other security features commonly used in both wireline and wireless applications.

The device implements three distinct VCAP units:

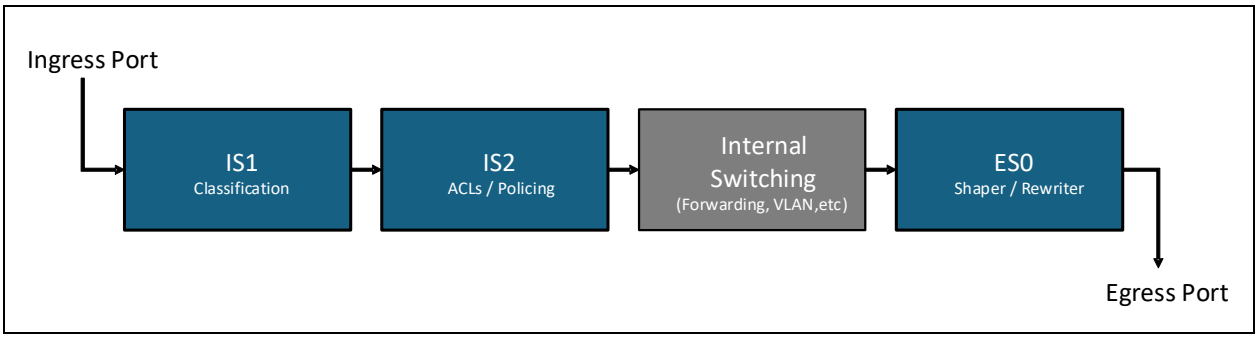
- **IS1** - Ingress Stage 1
- **IS2** - Ingress Stage 2
- **ES0** - Egress Stage 0

IS1 and IS2 operate on all incoming frames, performing classification, access control, and security checks. IS1 focuses on initial frame classification applying QoS, VLAN, and policer decisions based on frame headers and port configuration. IS2 builds on this by enforcing security policies such as access control, IP source guarding, and filtering rules based on deeper packet inspection.

ES0 operates on all outgoing frames and handles frame tagging and egress-specific manipulations.

At each stage (IS1, IS2, ES0) frames of interest can be identified by applying classification rules implemented in the switch hardware. Frames matching the configured classification criteria may then manipulated via selectable 'actions' that are also implemented in the switch hardware.

FIGURE 4-8: VCAP STAGES



4.10.1 VCAP PROCESSING OVERVIEW

For each VCAP processing stage that is enabled, frames undergo matching/classification based on their type (e.g., IPv4 TCP, IPv6 UDP). Specific fields are extracted from the frame and combined with port-specific configuration to form a **VCAP key instance**. This key instance is matched against entries in a ternary content-addressable memory (TCAM) with mask-based pattern matching. The first matching entry determines the corresponding action to apply.

Each entry in the VCAP comprises:

- A **pattern** (bit sequence to match)
- A **mask** (to define any "don't care" bits for matching)
- An **action set** (specifies how the packet should be treated if matched)

Each VCAP stage (IS0, IS1, ES0) is designed to support a particular type of function / use-case with each stage providing a specific set of available actions.

TABLE 4-16: VCAP STAGE ACTIONS

VCAP Stage	Action Type	Description / Use Case
IS1	Set QoS Class	Classify frame for queue selection / scheduling
	Set Drop Precedence (DP)	Mark for congestion-aware drop behavior
	Assign Policer Index	Apply ingress rate-limiting policy
	Set VLAN ID (VID)	Override ingress VLAN tagging
	DSCP Classification	Use DSCP for traffic class mapping
	Set Stream Identifier	Classify frame for stream handling
	Set PSFP Stream Filter Identifier	Classify frame for PSFP stream filter
	Set PSFP Stream Gate Identifier	Classify frame for PSFP stream gates
IS2	Permit / Drop	Allow or discard frames based on ACL match
	Trap to CPU	Forward frame to CPU for processing
	Mirror Frame	Send a copy to a mirror (SPAN) port
	MAC/IP Filtering Enforcement	Block specific L2/L3 addresses
	Apply Policer Index	Enforce security-related rate limits
	ACL Hit Counter Update	Track matches for statistics/logging
	Enable DoS/Source Guard Features	Mitigate attack surfaces (IP spoofing, flood detection, etc.)
ES0	VLAN Tag Insert/Modify/Remove	Egress VLAN manipulation (e.g., add/remove tags)
	Set Output VID	Rewrite VLAN ID on egress
	Set PCP / DEI	Set priority and drop eligibility bits
	Mark as Untagged	Suppress VLAN tag even if frame came in tagged

4.10.2 VCAP LOOKUP STAGES

Each VCAP stage inspects incoming or outgoing packets and makes decisions by comparing frame data against a set of selected predefined rules. This process is driven by two main components: a Key and an Entry.

4.10.2.1 Key

The key type is a predefined template notifying the hardware which fields to extract from a frame and how to order them into a binary representation for comparison against the TCAM entries. For example, a key type might instruct the hardware to extract and arrange in order fields such as:

- MAC source / destination address
- VLAN ID
- EtherType

When processing a frame the hardware generates an instance of the key. This contains the information (fields) extracted plus contextual information such as the ingress port and the general type of frame (e.g. IPv4 packet, IPv6 packet, etc). The key instance is then compared against the entry to determine if there is a match.

Specific key types are available in each VCAP stage (IS1, IS2, ES0). IS1 supports seven key types, IS2 supports thirteen key types, and ES0 supports one key type.

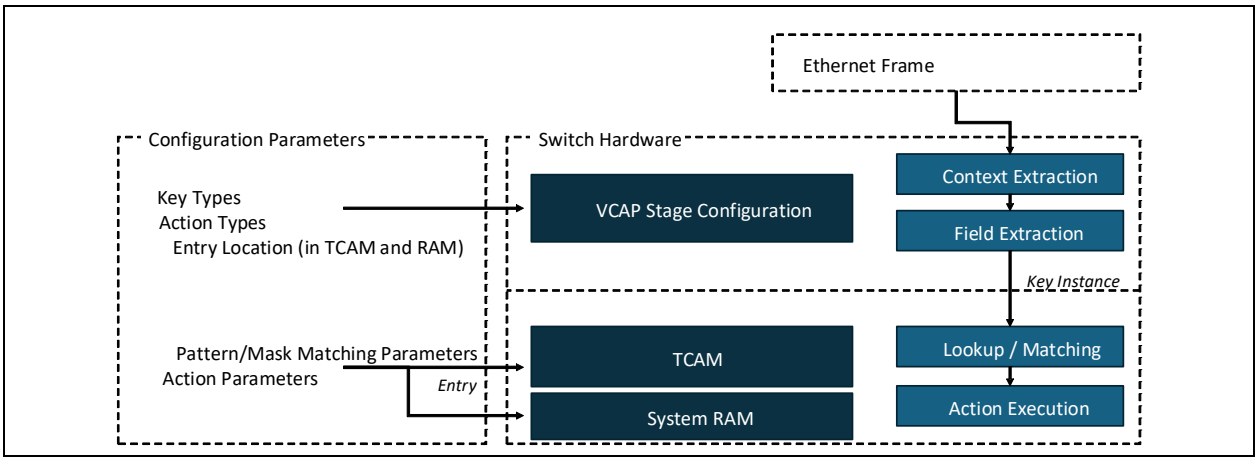
4.10.2.2 Entry

An Entry in the VCAP TCAM consists of:

- **Match pattern:** The values to compare against the Key instance
- **Mask:** To ignore certain fields during comparison ("don't care" bits)
- **Action:** What the switch should do if this entry matches

When a Key instance matches an Entry (based on the mask and pattern), the associated Action is executed (e.g. modifying frame headers, redirecting to the CPU, applying a QoS class, dropping the frame, etc.).

FIGURE 4-9: VCAP MATCHING



The TCAM provides sufficient storage for:

- 24 kB of IS1 entry data (mask and pattern) - supporting up to 64 to 256 entries depending upon the complexity of the key type / classification parameters
- 24 kB of IS2 entry data (mask and pattern) - supporting up to 64 to 256 entries
- 6 kB of ES0 entry data (mask and pattern) - supporting up to 64 entries

Each programmed entry may be associated with an action requiring storage in system RAM:

- 246-bits per IS1 action
- 62-bits per IS2 action
- 65-bits per ES0 action

4.10.3 LOOKUP OVERVIEW BY STAGE

Each frame can be subject to up to six ingress lookups (via IS1 and IS2) and one egress lookup per destination port (via ES0). Each lookup uses a specific Key format and matches against a specific type of Entry.

- Ingress Stage 1 (IS1): Advanced Classification
  - Up to three lookups can be performed in IS1 on any ingress packet packet a
  - IS1 is primarily responsible for advanced frame classification and policy enforcement
  - IS1 supports seven different key types allowing flexible classification of a wide variety of incoming traffic
- Ingress Stage 2 (IS2): Security and Filtering
  - Following IS1, up to three additional lookups can be performed in IS2
  - IS2 focuses on security enforcement, IP source guarding, and deep protocol inspection
  - IS2 supports 13 key types

4.10.4 PORT-BASED CONTROL OF VCAP BEHAVIOR

Each port in the device can influence how frames are evaluated by the VCAP system. This includes enabling or disabling lookups at different VCAP stages (IS1, IS2, ES0) and refining how packet fields are interpreted during key instance generation. These settings ensure that the VCAP pipeline can be flexibly adapted to the classification, filtering, or forwarding requirements of each port.

4.10.4.1 Ingress Stage 1 (IS1) - Classification Behavior per Port

Each port can individually control how lookups are performed. Key configuration options include:

- Enabling or disabling IS1 processing for incoming frames on a per-port basis.
- Choosing between source and destination fields. Each lookup can be configured to use either the source or destination MAC/IP addresses when generating the key instance.

- Selecting the VLAN tag context. When processing VLAN-tagged frames, the lookup can be set to use either the outer VLAN tag or the inner tag (for double-tagged frames).
- Choosing which key type to apply based on the type of traffic (e.g., IPv4, IPv6). Each key type defines a different set of packet fields to match against (such as source/destination addresses, ports, and protocol types).

4.10.4.2 Ingress Stage 2 (IS2) - Protocol Filtering and Access Control

IS2 can perform up to three additional lookups per ingress frame after IS1. Like IS1, IS2 behavior is configurable per port, with options such as:

- Enabling or disabling IS2 lookups for each port.
- Controlling frame-type matching logic: IS2 allows per-port control over how different protocols are interpreted and which entry types are used for matching. For example:
  - IPv6 frames can be matched using either deep (7-tuple) inspection or simpler matching templates.
  - IPv4 and ARP frames can be directed to specific lookup formats optimized for those protocols.
  - If desired, protocol-based matching can be bypassed entirely in favor of simpler Ethernet type-based matches.

Additionally, classification state from IS1 (such as the Policy Association Group or PAG) can influence how frames are handled in IS2, supporting multi-stage decision-making across ingress processing.

4.10.4.3 Egress Stage 0 (ES0) - Per-Port Output Policy

At egress, each port can optionally apply an ES0 lookup to modify or influence outgoing frames. When enabled, ES0 can apply actions such as:

- Modifying VLAN tags (insertion, removal, or rewriting).

4.10.5 ADVANCED VCAP FEATURES

4.10.5.1 Counter Capabilities

VCAP entries in IS2 can be configured with associated counters that increment on each match. These hit counters support real-time traffic monitoring and auditing of policy enforcement. They are especially useful for identifying active rules, tracking usage patterns, and detecting anomalous or unwanted flows.

4.10.5.2 Lookup Chaining: PAG and ISDX

VCAP enables inter-stage logic via chaining fields:

- **PAG (Policy Association Group):** Assigned during IS1 and used to guide IS2 decision-making, e.g., different ACLs for different traffic classes.
- **ISDX (Ingress Stream Index):** An index result from IS1 lookups that can influence IS2 and ES0 matching, allowing conditional logic between classification and access contr ol.

These fields act as metadata that bridge VCAP stages, allowing progressive classification and enforcement while avoiding redundant frame inspection.

4.11 Analyzer

The analyzer module is responsible for a number of tasks:

- Determining the set of destination ports, also known as the forwarding decision, for frames received by port modules. This includes Layer-2 forwarding, CPU-forwarding, mirroring, and sFlow sampling.
- Keeping track of network stations and their MAC addresses through MAC address learning and aging.
- Holding VLAN membership information (configured by CPU) and applying this to the forwarding decision.

The analyzer consists of the following main blocks.

- MAC table
- Parallelizable Media Access Controller (PMAC) table
- VLAN table
- Forwarding Engine

The MAC, PMAC, and VLAN tables are the main databases used by the forwarding engine. The forwarding engine determines the forwarding decision and initiates learning in the MAC table when appropriate.

The analyzer operates on analyzer requests initiated by the port modules. For each received frame, the port module requests the analyzer to determine the forwarding decision. Initially, the analyzer request is directed to the VCAP. The result from the VCAP (the IS2 action) is forwarded to the analyzer along with the original analyzer request. For more information about VCAP, see Section 4.10, VCAP.

The analyzer request contains the following frame information.

- Destination and source MAC addresses.
- Physical port number where the frame was received (referred to as PPORT).
- Logical port number where the frame was received (referred to as LPORT).
  By default, LPORT and PPORT are the same. However, when using link aggregation, multiple physical ports map to the same logical port. The LPORT value is configured in ANA:PORT:PORT_CFG.PORTID_VAL in the analyzer.
- Frame properties derived by the classifier and VCAP IS1:
  - Classified VID
  - Link aggregation code
  - Basic CPU forwarding
  - CPU forwarding for special frame types determined by the classifier

Based on this information, the analyzer determines an analyzer reply, which is returned to the ingress port modules. The analyzer reply contains:

- The forwarding decision (referred to as DEST). This mask contains 1 bit for each front port. DEST does not include the CPU port. The CPU port receives a copy of the frame if the CPU extraction queue mask, CPUQ, has any bits set.
- The CPU extraction queue mask (referred to as CPUQ). This mask contains 8 bits, 1 bit for each CPU extraction queue.

The terms PPORT, LPORT, DEST and CPUQ, as previously defined, are used throughout the remainder of this section.

4.11.1 MAC TABLE

This section provides information about the MAC table block in the analyzer.

The analyzer contains a MAC table with 8192 entries containing information about stations learned by the device. The table is organized as a hash table with four buckets and 2048 rows. Each row is indexed by an 11-bit hash value, which is calculated based on the station's (MAC, VID) pair, as shown in the following illustration.

FIGURE 4-10: MAC TABLE ORGANIZATION

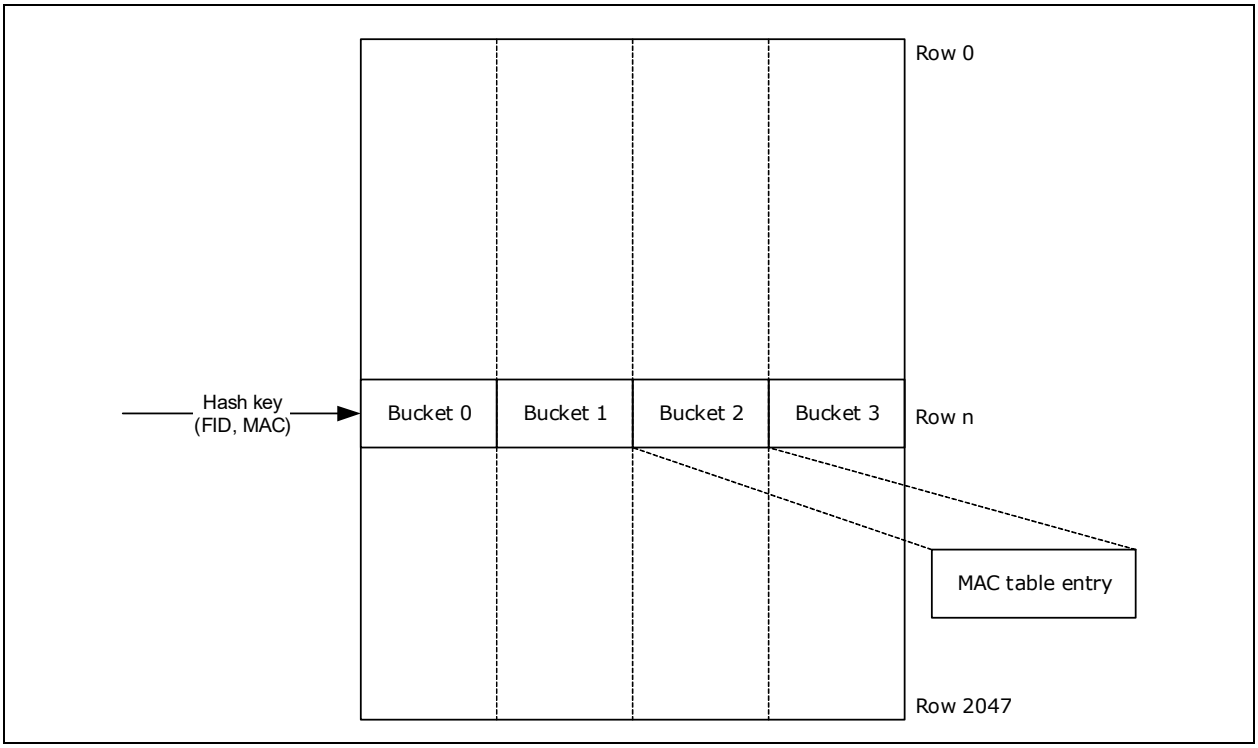


Table 4-17 lists the fields for each entry in the MAC table.

TABLE 4-17: MAC TABLE ENTRY

Field	Bits	Description
VALID	1	Entry is valid.
MAC	48	The MAC address of the station (primary key).
VID	13	VLAN identifier that the station is learned with (primary key). In reality, the MAC table stores a filtering identifier (FID). By default, the filtering identifier equals the VLAN identifier, which enables Independent VLAN Learning (IVL). See Section 4.11.1.8, Shared VLAN Learning for more details on the filtering identifier.
DEST_IDX	6	Destination mask index pointing to a destination mask in the destination mask table (PGID entries 0 through 63).
ENTRY_TYPE	2	Entry type: 0: Normal entry subject to aging. 1: Normal entry not subject to aging (locked). 2: IPv4 multicast entry not subject to aging. Full port set is encoded in MAC table entry. 3: IPv6 multicast entry not subject to aging. Full port set is encoded in MAC table entry.
AGED_FLAG	1	Entry is aged once by an age scan. See Section 4.11.1.2, Age Scan.
MAC_CPU_COPY	1	Copy frames from or to this station to the CPU.
SRC_KILL	1	Do not forward frames from this station. This flag is not used for destination lookups.
IGNORE_VLAN	1	Do not use the VLAN_PORT_MASK from the VLAN table when forwarding frames to this station.

TABLE 4-17: MAC TABLE ENTRY (CONTINUED)

Field	Bits	Description
CHANGE2SW	1	Indicates to software that an entry was changed since the last SYNC-GET_NEXT command was run.

Entries in the MAC table can be added, deleted, or updated in three ways:

- Hardware-based learning of source MAC addresses (that is, inserting new (MAC, VID) pairs in the MAC table).
- Age scans (setting AGED_FLAG and deleting entries).
- CPU commands (for example, for CPU-based learning).

4.11.1.1 Hardware-Based Learning

The analyzer adds an entry to the MAC table when learning is enabled, and the MAC table does not contain an entry for a received frame's (SMAC, VID). The new entry is formatted as follows:

- VALID is set.
- MAC is set to the frame's SMAC.
- VID is set to the frame's VID.
- ENTRY_TYPE is set to 0 (normal entry subject to aging).
- DEST_IDX is set to the frame's LPORT.
- MAC_CPU_COPY is set to AGENCTRL.LEARN_CPU_COPY.
- SRC_KILL is set to AGENCTRL.LEARN_SRC_KILL.
- IGNORE_VLAN is set to AGENCTRL.LEARN_IGNORE_VLAN.
- All other fields are cleared.

When a frame is received from a known station, that is, the MAC table already contains an entry for the received frame's (SMAC, VID), the analyzer can update the entry as follows.

For entries of entry type 0 (unlocked entries):

- The AGED_FLAG is cleared. This implies the station is active, avoiding the deletion of the entry due to aging.
- If the existing entry's DEST_IDX differs from the frame's LPORT, then the entry's DEST_IDX is set to the frame's LPORT. This implies the station has moved to a new port.

For entries of entry type 1 (locked entries):

- The AGED_FLAG is cleared. This implies the station is active.

Entries of entry types 2 and 3 are never updated, because their multicast MAC addresses are never used as source MAC addresses.

For more information about learning, see Section 4.11.4.5, SMAC Analysis.

4.11.1.2 Age Scan

The analyzer scans the MAC table for inactive entries. An age scan is initiated by either a CPU command or automatically performed by the device with a configurable age scan period (AUTOAGE). The age scan checks the flag AGED_FLAG for all entries in the MAC table. If an entry's AGED_FLAG is already set and the entry is of entry type 0, the entry is removed. If the AGED_FLAG is not set, it is set to 1. The flag is cleared when receiving frames from the station identified by the MAC table entry. For more information, see Section 4.11.1.1, Hardware-Based Learning.

4.11.1.3 CPU Commands

Table 4-18 lists the set of commands that a CPU can use to access the MAC table. The MAC table command is written to MACACCESS.MAC_TABLE_CMD. Some commands require the registers MACLDATA, MACHDATA, and MACTINDX to be preloaded before the command is issued. Some commands return information in MACACCESS, MACLDATA, and MACHDATA.

TABLE 4-18: MAC TABLE COMMANDS

Command	Purpose	Use
LEARN	Insert/learn new entry in MAC table. Position given by (MAC, VID)	Configure MAC and VID of the new entry in MACHDATA and MACLDATA. Configure remaining entry fields in MACACCESS. The location in the MAC table is calculated based on (MAC, VID).
FORGET	Delete/unlearn entry given by (MAC, VID).	Configure MAC and VID in MACHDATA and MACLDATA.
AGE	Start age scan.	No preload required. Issue command.
READ	Read entry pointed to by (row, column).	Configure row (0-2047) and column (0-3) of the entry to read in: MACTINDX.INDEX (row). MACTINDX.BUCKET (column). MACACCESS.VALID must be set to 0. When MAC_TABLE_CMD changes to IDLE, MACHDATA, MACLDATA, and MACACCESS contain the information read.
LOOKUP	Lookup entry pointed to by (MAC, VID).	Configure MAC and VID of station to look up in MACHDATA and MACLDATA. MACACCESS.VALID must be 1. Issue a READ command. When MAC_TABLE_CMD changes to IDLE, success of the lookup is indicated by MACACESS.VALID. If successful, MACACCESS contains the entry information.
WRITE	Write entry, MAC table position given by (row, column).	Configure MAC and VID of the new entry in MACHDATA and MACLDATA. Configure remaining entry fields in MACACCESS. The location in the MAC table is given by row and column in MACTINDX.
INIT	Initialize the table.	No preload required. Issue command.
GET_NEXT	Get the smallest entry in the MAC table numerically larger than the specified (MAC, VID). The VID and MAC are evaluated as a 60-bit number with the VID being most significant.	Configure MAC and VID of the starting point for the search in MACHDATA and MACLDATA. When MAC_TABLE_CMD changes to IDLE, success of the search is indicated by MACACESS.VALID. If successful, MACHDATA, MACLDATA, and MACACCESS contain the information read.
SYNC-GET_NEXT	Get the next entry in the MAC table where the CHANCE2SW flag is set.	Configure row (0-2047) and column (0-3) of the starting point for the search in MACTINDX.INDEX (row) and MACTINDX.BUCKET (column). When MAC_TABLE_CMD changes to IDLE, MACHDATA, MACLDATA, and MACACCESS contain the information read.
IDLE	Indicate that MAC table is ready for new command.	No preload required.

4.11.1.4 Known Multicasts

From a CPU, entries can be added to the MAC table with any content. This makes it possible to add a known multicast address with multiple destination ports:

- Set the MAC and VID in MACHDATA and MACLDATA.
- Set MACACCESS.ENTRY_TYPE = 1, because this is not an entry subject to aging.
- Set MACACCESS.AGED_FLAG to 0.
- Set MACACCESS.DEST_IDX to an unused value.

- Set the destination mask in the destination mask table pointed to by DEST_IDX to the desired ports.

Example All frames in VLAN 12 with MAC address 0x010000112233 are to be forwarded to ports 8, 9, and 12.

This is done by inserting the following entry in the MAC table:

VID = 12
MAC = 0x010000112233
ENTRY_TYPE = 1
VALID = 1
AGED_FLAG = 0
DEST_IDX = 40

and configuring the destination mask table PGID[40] = 0x1300.

IPv4 and IPv6 multicast entries can be programmed differently without using the destination mask table. This is described in the following subsection.

4.11.1.5 IPv4 Multicast Entries

MAC table entries with the ENTRY_TYPE = 2 settings are interpreted as IPv4 multicast entries.

IPv4 multicasts entries match IPv4 frames, which are classified to the specified VID, and which have DMAC = 0x01005Exxxxxx, where xxxxxx is the lower 24 bits of the MAC address in the entry.

Instead of a lookup in the destination mask table (PGID), the destination set is programmed as part of the entry MAC address. This is shown in Table 4-19.

TABLE 4-19: IPV4 MULTICAST DESTINATION MASK

Destination Ports	Record Bit Field
Ports 7-0	MAC[31-24]

Example: All IPv4 multicast frames in VLAN 12 with MAC 01005E112233 are to be forwarded to ports 3, 4, and 7. This is done by inserting the following entry in the MAC table entry:

VALID = 1
VID = 12
MAC = 0x000098112233
ENTRY_TYPE = 2
DEST_IDX = 0

4.11.1.6 IPv6 Multicast Entries

MAC table entries with the ENTRY_TYPE = 3 settings are interpreted as IPv6 multicast entries. IPv6 multicasts entries match IPv6 frames, which are classified to the specified VID, and which have DMAC=0x3333xxxxxxxx, where xxxxxxxx is the lower 32 bits of the MAC address in the entry.

Instead of a lookup in the destination mask table (PGID), the destination set is programmed as part of the entry MAC address. This is shown in Table 4-20.

TABLE 4-20: IPV6 MULTICAST DESTINATION MASK

Destination Ports	Record Bit Field
Ports 7-0	MAC [39-32]

Example: All IPv6 multicast frames in VLAN 12 with MAC 333300112233 are to be forwarded to ports 3, 4, and 7.

This is done by inserting the following entry in the MAC table entry:

VID = 12
MAC = 0x009800112233
ENTRY_TYPE = 3
VALID = 1
DEST_IDX = 0

4.11.1.7 Port and VLAN Filter

The ANAGEFIL register can be used to only hit specific ports or VLANs when doing certain operations. If the filter is enabled, it affects the Manual age scan command (MACACCESS.MAC_TABLE_CMD = AGE).

The GET_NEXT MAC table command. For more information, see Section 4.11.1.3, CPU Commands.

When both filters are enabled at the same time, all conditions must be fulfilled before the operation (aging, GET_NEXT) is carried out.

4.11.1.8 Shared VLAN Learning

In the default configuration, the device is set up to do Independent VLAN Learning (IVL), that is, MAC addresses are learned separately on each VLAN. The device also supports Shared VLAN Learning (SVL), where a MAC table entry is shared among a group of VLANs. For shared VLAN learning, a MAC address and a Filter Identifier (FID) define each MAC table entry. A set of VIDs then map to the FID.

The device supports shared VLAN learning in three ways:

- Through the IS1 actions FID_SEL and FID_VAL specifying the FID to use.
- Through the per-VID mapping table FID_MAP.
- Through the AGENCTRL.FID_MASK, which controls a general mapping between FID and VIDs.

The IS1 action FID_SEL selects whether to use the FID_VAL for the DMAC lookup, for the SMAC lookup (learning), or for both lookups. If set for a lookup, the FID_VAL replaces the VID when calculating the hash key into the MAC table and when comparing with the entry's VID. If used during the SMAC lookup, new entries are learned using the FID_VAL. If an IS1 action returns a FID_SEL > 0, it overrules the use of the FID mapping table for the frame. In addition, FID_SEL > 0 overrules the use of the FID_MASK for the MAC table lookups specified in FID_SEL.

The FID mapping table, FID_MAP, maps the frame's classified VID to a FID. If the returned FID is larger than 0, then the FID overrules the use of the FID_MASK for both the DMAC and SMAC lookups in the MAC table. Learning is done using the returned FID.

If neither IS1 nor the FID_MAP table have instructed changes to the FID, then the FID_MASK is applied. The 12-bit FID_MASK masks out the corresponding bits in the VID. The FID used for learning and lookup is therefore calculated as FID = VID AND (NOT FID_MASK). The FID used in the MAC table is 13 bits so the calculated FID is prepended with 0. The most significant bit in the FID in the MAC table is only selectable through the FID_ENA action out of IS1.

All VIDs mapping to the same FID share the same MAC table entries.

Example: Configure all MAC table entries to be shared among all VLANs.

This is done by setting FID_MASK to 111111111111.

Example: Split the MAC table into two separate databases: one for even VIDs and one for odd VIDs.

This is done by setting FID_MASK to 111111111110.

4.11.1.9 Learn Limit

The ENTRYLIM.ENTRYLIM register specifies the maximum number of unlocked entries in the MAC table that a port is allowed to use. Locked and IPMC entries are not taken into account.

After the limit is reached, both auto-learning and CPU-based learning on unlocked entries are denied. A learn frame causing the limit to be exceeded can be copied to the CPU (PORT_CFG.LIMIT_CPU) and the forwarding to other front ports can be denied (PORT_CFG.LIMIT_DROP).

The ENTRYLIM.ENTRYSTAT register holds the current number of entries in the MAC table. MAC table aging and manual removing of entries through the CPU cause the current number to be reduced. If a MAC table entry moves from one port to another port, this is also reduces the current number. If the move causes the new port's limit to be exceeded, the entry is denied and removed from the MAC table.

The LEARNDISC counts all events where a MAC table entry is not created or updated due to a learn limit.

4.11.2 PMAC TABLE

The analyzer contains a PMAC table with 8,192 entries containing information about stations learned by the device. The intended use of the PMAC table is for Profinet streams using an optimized index model for guaranteed MAC table access without the potential collisions associated with hashed MAC tables. The PMAC table is organized as 2,048 entries per VLAN supporting up to 4 VLANs. The PMAC table is enabled in ANA::PMAC_CFG.PMAC_ENA.

The PMAC table is looked up for destination MAC addresses only. Frames subject to a PMAC table lookup must fulfill the following criteria:

- OUI from frame's destination MAC address must match the configured value in ANA::PMAC_CFG and ANA::PMAC_CFG_2.
- Frame's classified VLAN must match one of the configured values in ANA::PMAC_VLAN_CFG.

The PMAC table is looked up using the following index:

- DMAC[10:0] + 2,048*vlan_match_idx,

where the vlan_match_idx is the match index into ANA::PMAC_VLAN_CFG.

If a frame is subject to a PMAC table lookup and the result of the lookup is a valid PMAC entry then the PMAC entry replaces the result of the DMAC lookup in the normal MAC table. Otherwise, the normal MAC table is used for forwarding. The SMAC lookup in the normal MAC table is always performed, independently of whether the PMAC table is in use.

Table 4-21 lists the fields for each entry in the PMAC table.

TABLE 4-21: FIELDS IN THE PMAC TABLE

Field	Bits	Description
VALID	1	Entry is valid.
DEST_IDX	6	Destination mask index pointing to a destination mask in the destination mask table (PGID entries 0 through 63).
MAC_CPU_COPY	1	Copy frames to this station to the CPU.
IGNORE_VLAN	1	Do not use the VLAN_PORT_MASK from the VLAN table when forwarding frames to this station.

In contrast to the MAC table, the PMAC table is static. This means all entries are added, deleted, and updated by CPU access only.

Table 4-22 lists the set of commands that a CPU can issue to access the PMAC table. The PMAC table command is written to PMACACCESS.PMAC_TBL_CMD.

TABLE 4-22: PMAC TABLE COMMANDS

Command	Purpose	Use
INIT	Initialize the table.	Issue command. When PMAC_TBL_CMD changes to IDLE, initialization has completed and all entries are cleared.
READ	Read PMAC table entry for specific index.	Configure the index to read from in PMACTIDX.PMAC_INDEX. When PMAC_TBL_CMD changes to IDLE, PMACACCESS contains the information read.
WRITE	Write PMAC table entry for specific index.	Configure the index to write to in PMACTIDX.PMAC_INDEX. Configure the content of the PMAC entry in: PMACACCESS.PMAC_VALID, PMACACCESS.PMAC_IGNORE_VLAN, PMACACCESS.PMAC_CPU_COPY, PMACACCESS.PMAC_DEST_IDX.
IDLE	Indicate that PMAC table is ready for new command.	No preload required.

4.11.3 VLAN TABLE

The analyzer has a VLAN table that contains information about the members of each of the 4,096 VLANs. Table 4-23 lists fields for each entry in the VLAN table.

TABLE 4-23: FIELDS IN THE VLAN TABLE

Field	Bits	Description
VLAN_PORT_MASK	6	One bit for each port. Set if port is member of VLAN. The CPU port is always a member of all VLANs.

TABLE 4-23: FIELDS IN THE VLAN TABLE (CONTINUED)

Field	Bits	Description
VLAN_MIRROR	1	Mirror frames received in the VLAN. See Section 4.11.4.7, Mirroring.
VLAN_SRC_CHK	1	VLAN ingress filtering. If set, frames classified to this VLAN are dropped if PPORT is not member of the VLAN.
VLAN_LEARN_DISABLED	1	Disable learning in the VLAN.
VLAN_PRIV_VLAN	1	Set VLAN to private.
VLAN_PGID_CPU_DIS	1	Disable a CPU-copy based on the DMAC-based lookup in the PGID table. See Section 4.11.4.1, DMAC Analysis.

By default, all ports are members of all VLANs. This default can be changed through a CPU command. Table 4-24 lists the set of commands that a CPU can issue to access the VLAN table. The VLAN table command is written to VLANACCESS.VLAN_TBL_CMD.

TABLE 4-24: VLAN TABLE COMMANDS

Command	Purpose	Use
INIT	Initialize the table	Issue command. When VLAN_TBL_CMD changes to IDLE, initialization has completed and all ports are member of all VLANs. All flags are cleared.
READ	Read VLAN table entry for specific VID.	Configure the VLAN to read from in VLANTIDX.V_INDEX. When VLAN_TBL_CMD changes to IDLE, VLANACCESS, and VLANTIDX contain the information read.
WRITE	Write VLAN table entry for specific VID.	Configure the VLAN to write to in VLANTIDX.V_INDEX. Configure the content of the VLAN record in: VLANACCESS.VLAN_PORT_MASK, VLANTIDX.VLAN_MIRROR, VLANTIDX.VLAN_SRC_CHK, VLANTIDX.VLAN_LEARN_DISABLED, VLANTIDX.VLAN_PRIV_VLAN, VLANTIDX.VLAN_PGID_CPU_DIS.
IDLE	Indicate that VLAN table is ready for new command	No preload required.

4.11.4 FORWARDING ENGINE

The analyzer determines the set of ports to which each frame is forwarded, in several configurable steps. The resulting destination port set can include any number of front ports, excluding the CPU port. The CPU port is handled through the CPU extraction queue mask.

The analyzer request from the port modules is passed through all the processing steps of the forwarding engine. As each step is carried out, the destination port set (DEST) and CPU extraction queue mask (CPUQ) are built up.

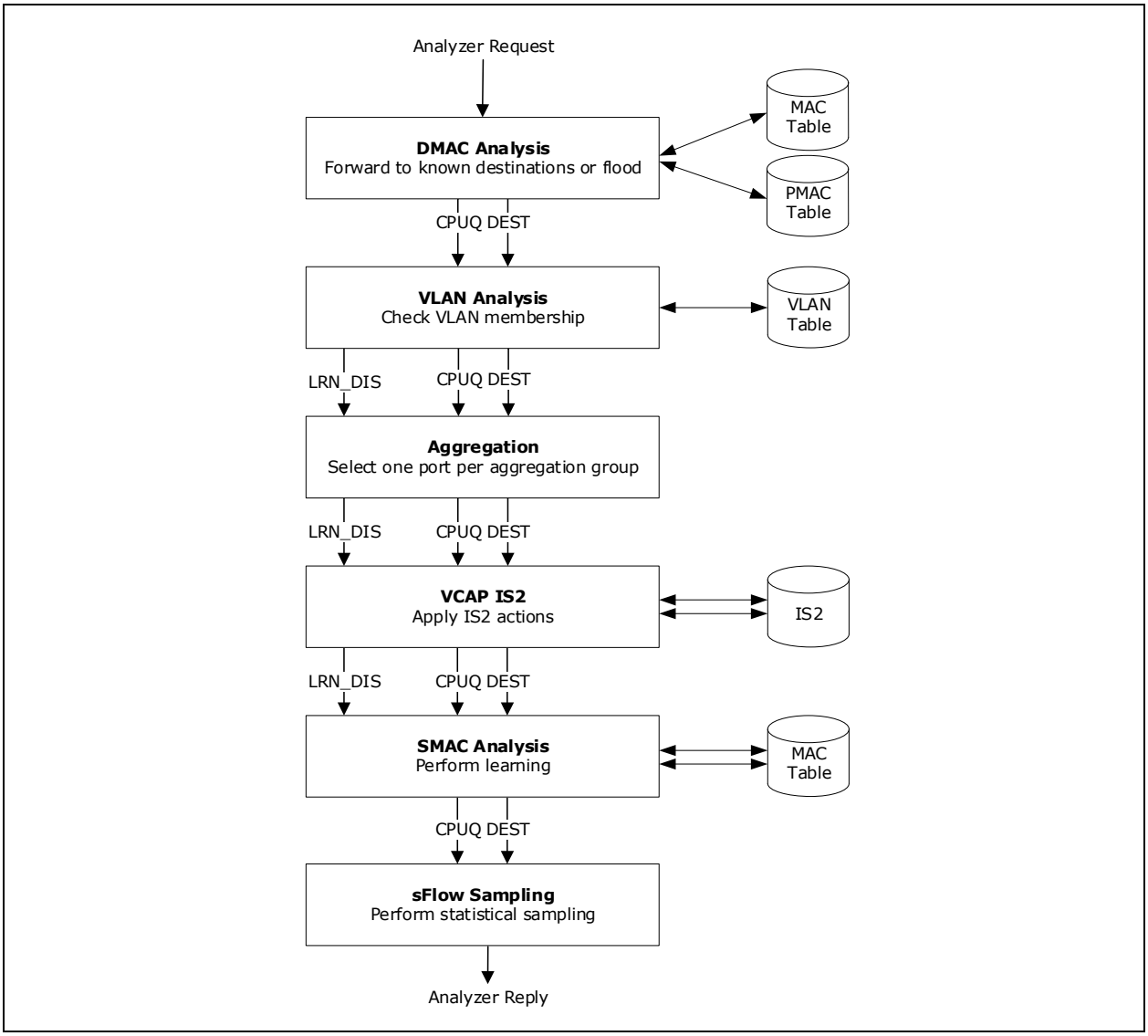
In addition to the forwarding decision, the analyzer determines which frames are subject to learning (also known as learn frames). Learn frames trigger insertion of a new entry in the MAC table or update of an existing entry. Learning is presented as part of the forwarding, because in some cases, learning changes the normal forwarding of a frame, such as secure learning.

During the processing, the analyzer determines a local frame property. The learning-disabled flag, LRN_DIS is used in the SMAC Learning step:

- If the learning-disabled flag is set, learning based on (SMAC, VID) is disabled.
- If the learning-disabled flag is cleared, learning is conducted according to the configuration in the SMAC learning step.

Figure 4-11 shows the configuration steps in the analyzer.

FIGURE 4-11: ANALYSIS STEPS



4.11.4.1 DMAC Analysis

During the DMAC analysis step, the PMAC table and the MAC table are looked up to get the first input to the calculation of the destination port set. For more information about the MAC table, see Section 4.11.1, MAC Table. For more information about the PMAC table, see Section 4.11.2, PMAC Table.

For frames subject to a PMAC table lookup (VID matching on of the PMAC VIDs and OUI of DMAC matching the PMAC OUI), the (DMAC, VID) pair provides an index to the PMAC table. A match is found when the indexed entry is valid.

Next, the (DMAC, VID) pair is looked up in the MAC table. A match is found when the (DMAC, VID) pair matches that of an entry.

A match in the PMAC table takes precedence over a match in the MAC table. If a match is found in the PMAC table, the PMAC table entry is returned and DEST is determined based on that entry. Otherwise, if a match is found in the MAC table, the MAC table entry is returned and DEST is determined based on that entry instead.

If an entry is found, the flag MAC_CPU_COPY is processed. If MAC_CPU_COPY is set, the CPUQ_CFG.CPUQ_MAC is added to CPUQ. The processing of this flag can be disabled through AGENCTRL.IGNORE_DMAC_FLAGS.

If an entry is not found for the (DMAC, VID) in neither the PMAC table nor the MAC table, the frame is flooded. The forwarding decision is set to one of the flooding masks defined in ANA::FLOODING or ANA::FLOODING_IPMC, based on one of the flood type definitions listed in Table 4-25.

TABLE 4-25: FORWARDING DECISIONS BASED ON FLOOD TYPE

Frame Type	Condition	Replication
IPv4 multicast data	DMAC = 0x01005E000000 to 0x01005E7FFFFF EtherType = IPv4 IP protocol is not IGMP IPv4 DIP outside 224.0.0.x	None
IPv6 multicast data	DMAC = 0x333300000000 to 0x3333FFFFFFFF EtherType = IPv6 IPv6 DIP outside 0xFF02::/16	None
IPv4 multicast control	DMAC = 0x01005E000000 to 0x01005E7FFFFF EtherType = IPv4 IP protocol is not IGMP IPv4 DIP inside 224.0.0.x	None
IPv6 multicast control	DMAC = 0x333300000000 to 0x3333FFFFFFFF EtherType = IPv6 IPv6 DIP inside 0xFF02::/16	None
Broadcast	DMAC = 0xFFFFFFFFFFFF non-IPv4-multicast-data non-IPv6-multicast-data non-IPv4-multicast-control non-IPv6-multicast-control	Per QoS class
Multicast	Bit 40 in DMAC = 1 non-broadcast non-IPv4-multicast-data non-IPv6-multicast-data non-IPv4-multicast-control non-IPv6-multicast-control	Per QoS class
Unicast	Bit 40 in DMAC = 0	Per QoS class

Next, CPU-forwarding from the basic classifier, and the IS2 SMAC_SIP lookup is processed if:

- The basic classifier decided to copy the frame to the CPU, the corresponding CPU extraction queue is added to CPUQ.
- The basic classifier decided to redirect the frame to the CPU, DEST is cleared and the corresponding CPU extraction queue is added to CPUQ.
- The IS2 SMAC_SIP result decided to copy the frame to the CPU (SMAC_SIP_ACTION.CPU_COPY_ENA), the corresponding CPU extraction queue, SMAC_SIP_ACTION.CPU_QU_NUM is added to CPUQ.
- The IS2 SMAC_SIP result decided to discard the frame (SMAC_SIP_ACTION.FWD_KILL_ENA set), DEST is cleared.

For more information about frame type definitions for CPU forwarding in the basic classifier, see Table 4-15.

4.11.4.2 VLAN Analysis

During the VLAN analysis step, VLAN configuration is taken into account. As a result, ports can be removed from the forwarding decision. For more information about VLAN configuration, see Section 4.11.3, VLAN Table.

The frame's VID is used as an address for lookup in the VLAN table and the returned VLAN information is processed as follows:

- All ports that are not members of the VLAN are removed from DEST, except if the (DMAC, VID) match in the PMAC or MAC table has VLAN_IGNORE set, or if there is no match in neither the PMAC nor the MAC table and AGENCTRL.FLOOD_IGNORE_VLAN is set.

  Note These two exceptions are skipped if AGENCTRL.IGNORE_DMAC_FLAGS is set.
- If the VLAN_PRIV_VLAN flag in the VLAN table is set, the VLAN is private, and isolated and community ports

must be treated differently. An isolated port is identified as an ingress port for which PPORT is cleared in the ISOLATED_PORTS register. An community port is identified as an ingress port for which PPORT is cleared in the COMMUNITY_PORTS register. For frames received on an isolated port, all isolated and community ports are removed from the forwarding decision. For frames received on a community port, all isolated ports are removed from the forwarding decision.

- If VLAN ingress filtering is enabled, it is checked whether PPORT is member of the VLAN (VLAN_PORT_MASK). If this is not the case, DEST is cleared. VLAN ingress filtering is enabled per port in the VLANMASK register or per VLAN in the VLAN_SRC_CHK flag in the VLAN table. If either is set, VLAN ingress filtering is performed.

Next, it is checked whether the ingress port is enabled to forward frames to other front ports and the source mask (PGID[80+PPORT]) is processed as follows:

- If PORT_CFG.RECV_ENA for PPORT is 0, DEST is cleared.
- Any ports, which are cleared in PGID[80+PPORT], are removed from DEST.

SMAC learning is disabled by setting the LRN_DIS flag when either of the following two conditions is fulfilled as follows:

- VLAN_LEARN_DISABLED is set in the VLAN table for the VLAN.
- A frame is subject to VLAN ingress filtering (frame dropped due to PPORT not being member of VLAN), and ADV_LEARN.VLAN_CHK is set.

Finally, if the (DMAC, VID) match an entry in the PMAC table or an entry in the MAC table of ENTRY_TYPE 0 or 1 and the CPU port is set in the PGID pointed to by the entry, CPU extraction queue PGID.CPUQ_DST_PGID is added to the CPUQ when either of the following two conditions is fulfilled as follows:

- VLAN_PGID_CPU_DIS is cleared in the VLAN table for the VLAN.
- PGID.OBEY_VLAN is cleared for the destination PGID used for forwarding.

4.11.4.3 Aggregation

During the aggregation step, link aggregation is handled.

The aggregation step ensures that when a frame is destined for an aggregation group, it is forwarded to exactly one of the group's member ports.

For non-aggregated ports, there is a one-to-one correspondence between logical port (LPORT) and physical port (PPORT). The aggregation step does not change the forwarding decision.

For aggregated ports, all physical ports in the aggregation group map to the same logical port, and the entry in the destination mask table for the logical port includes all physical ports, which are members of the aggregation group. As a result, all but one member port must be removed from the destination port set.

The link aggregation code generated in the classifier is used to look up an aggregation mask in the aggregation masks table. Finally, ports that are cleared in the selected aggregation mask are removed from DEST.

4.11.4.4 VCAP Action Handling

VCAP IS2 actions are processed during the VCAP IS2 action handling step. Table 4-26 lists the processing of the VCAP actions. The order of processing is from top to bottom.

TABLE 4-26: VCAP IS2 ACTION PROCESSING

IS2 Action Field	Description
CPU_COPY_ENA CPU_QU_NUM	If CPU_COPY_ENA is set, the CPU_QU_NUM bit is set in CPUQ.
HIT_ME_ONCE CPU_QU_NUM	If HIT_ME_ONCE is set and the HIT_CNT counter is zero, the CPU_QU_NUM bit is set in CPUQ.
LRN_DIS	If set, learning is disabled (LRN_DIS flag is set).
POLICE_ENA POLICE_IDX	If POLICE_ENA is set (only applies to first lookup), the POLICE_IDX instructs which policer to use for this frame. See Section 4.12, Policers.
POLICE_VCAP_ONLY	If POLICE_VCAP_ONLY is set (only applies to first lookup), the only active policer for this frame is the VCAP policer. Other policers (QoS, port, PSFP) are disabled. See Section 4.12, Policers.

TABLE 4-26: VCAP IS2 ACTION PROCESSING (CONTINUED)

IS2 Action Field	Description
MASK_MODE PORT_MASK	The following actions are defined for MASK_MODE. 0: No action. 1: Permit. Ports cleared in PORT_MASK are removed from DEST. 2: Policy. DEST from the DMAC analysis step is replaced with PORT_MASK 3: Redirect. DEST as the outcome of the DMAC, VLAN, service, and aggregation analysis steps is replaced with PORT_MASK.
MIRROR_ENA	If MIRROR_ENA is set, mirroring is enabled. This is used in the mirroring step. See Section 4.11.4.7, Mirroring.

4.11.4.5 SMAC Analysis

During the SMAC analysis step, the MAC table is searched for a match against the (SMAC, VID), and the MAC table is updated due to learning. The learning part is skipped if the LRN_DIS flag was set by any of the previous steps.

Three different type of learn frames are identified:

- **Normal learn frames**. Frames for which an entry for the (SMAC, VID) is not found in the MAC table or the (SMAC, VID) entry in the MAC table is unlocked and has a DEST_IDX different from LPORT. In addition, the learn limit for the LPORT must not be exceeded (ENTRYLIM).
- **Learn frames exceeding the learn limit**. Same condition as for normal learn frames except that the learn limit for the LPORT is exceeded (ENTRYLIM)
- **Learn frames triggering a port move of a locked MAC table entry**. Frames for which the (SMAC, VID) entry in the MAC table is locked and has a DEST_IDX different from LPORT.

For all learn frames, the following must apply before learning related processing is applied.

- Learning is enabled by PORT_CFG.LEARN_ENA.
- The LRN_DIS flag from previous processing steps must be cleared, which implies the following:
  Learning is not disabled due to VLAN ingress filtering
  Learning is not disabled due to VCAP IS2 action
  Learning is enabled for the VLAN (VLAN_LEARN_DISABLED is cleared in the VLAN table)

In addition, learning must not be disabled due to the ingress policer having policed the frame. For more information, see Section 4.12, Policers.

If learning is enabled, learn frames are processed according to the setting of the following configuration parameters.

4.11.4.5.1 Normal Learn Frames

- Automatic learning. If PORT_CFG.LEARNAUTO is set for PPORT, the (SMAC, VID) entry is automatically added to the MAC table in the domain being searched.
- Drop learn frames. If PORT_CFG.LEARNDROP is set for PPORT, DEST is cleared for learn frames. Therefore, learn frames are not forwarded on any ports. This is used for secure learning, where the CPU must verify a station before forwarding is allowed.
- Copy learn frames to the CPU. If PORT_CFG.LEARNCPU is set for PPORT, the CPU port is added to DEST for learn frames and CPUQ_CFG.CPUQ_LRN is set in CPUQ. This is used for CPU based learning.

4.11.4.5.2 Learn Frames Exceeding the Learn Limit

- Drop learn frames. If PORT_CFG.LIMIT_DROP is set for PPORT, DEST is cleared for learn frames. As a result, learn frames are not forwarded on any ports.
- Copy learn frames to the CPU – If PORT_CFG.LIMIT_CPU is set for PPORT, the CPU port is added to DEST and CPUQ_CFG.CPUQ_LRN is set in CPUQ for learn frames.

4.11.4.5.3 Learn Frames Triggering a Port Move of a Locked MAC Table Entry

- Drop learn frames. If PORT_CFG.LOCKED_PORTMOVE_DROP is set for PPORT, DEST is cleared for learn frames. Therefore, learn frames are not forwarded on any ports.
- Copy learn frames to the CPU. If PORT_CFG.LOCKED_PORTMOVE_CPU is set for PPORT, the CPU port is added to DEST and CPUQ_CFG.CPUQ_LOCKED_PORTMOVE is added to CPUQ.

Finally, if a match is found in the MAC table for the (SMAC, VID), adjustments can be made to the forwarding decision.

- If the (SMAC, VID) match in the MAC table has SRC_KILL set, DEST is cleared.
- If the (SMAC, VID) match in the MAC table has MAC_CPU_COPY set, CPUQ_CFG.CPUQ_MAC_COPY is added to CPUQ.

The processing of the MAC table flags from the (SMAC, VID) match can be disabled through AGENCTRL.IGNORE_SMAC_FLAGS.

4.11.4.6 sFlow Sampling

This process step handles sFlow sampling.

sFlow is a standard for monitoring high-speed switch networks through statistical sampling of incoming and outgoing frames. Each port in the device can be setup as an sFlow agent monitoring the particular link and generating sFlow data. If a frame is sFlow sampled, it is copied to the sFlow CPU extraction queue (CPUQ_SFLOW).

An sFlow agent is configured through SFLOW_CFG with the following options:

- SF_RATE specifies the probability that the sampler copies a frame to the CPU. Each frame being candidate for the sampler has the same probability of being sampled. The rate is set in steps of 1/4096.
- SF_SAMPLE_RX enables incoming frames on the port as candidates for the sampler.
- SF_SAMPLE_TX enables outgoing frames on the port as candidates for the sampler.

The Rx and Tx can be enabled independently. If both are enabled, all incoming and outgoing traffic on the port is subject to the statistical sampling given by the rate in SF_RATE.

4.11.4.7 Mirroring

Mirroring is a processing step that replicates selected traffic to a designated destination port while not affecting or interrupting the normal routing of the traffic.

Frames subject to mirroring are identified based on the following mirror probes:

- Learn mirroring if ADVLEARN.LEARN_MIRROR is set and frame is a learn frame.
- CPU mirroring if AGENCTRL.MIRROR_CPU is set and the CPU port is set in DEST.
- Ingress mirroring if PORT_CFG.SRC_MIRROR_ENA is set.
- Egress mirroring if any port set in EMIRRORMASK is also set in DEST.
- VLAN mirroring if VLAN_MIRROR set in the VLAN table entry.
- VCAP mirroring if an action is hit that requires mirroring.

The following adjustment is made to the forwarding decision for frames subject to mirroring:

- Ports set in MIRRORPORTS are added to DEST.

If the CPU port is set in the MIRRORPORTS, CPU extraction queue CPUQ_CFG.CPUQ_MIRROR is added to the CPUQ.

For learn frames with learning enabled, all ports in ADVLEARN.LEARN_MIRROR are added to DEST. For more information, see Section 4.11.4.5, SMAC Analysis.

4.11.5 ANALYZER MONITORING

Miscellaneous events in the analyzer can be monitored, which can provide an understanding of the events during the processing steps.

Port moves, defined as a known station moving to a new port, are registered in the ANMOVED register. A port move occurs when an existing MAC table entry for (MAC, VID) is updated with new port information (DEST_IDX). Such an event is registered in ANMOVED by setting the bit corresponding to the new port.

Continuously occurring port moves may indicate a loop in the network or a faulty link aggregation configuration.

A list of events, such as frame flooding or policer drop, can be monitored in ANEVENTS and ANEVENTS2.

The LEARNDISC counter registers every time an entry in the MAC table cannot be made or if an entry is removed due to lack of storage.

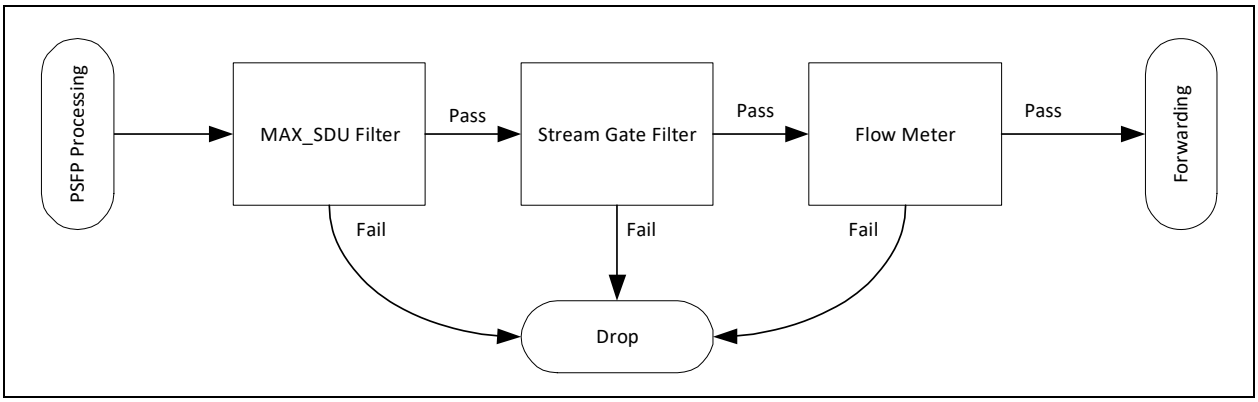
4.11.6 PER-STREAM FILTERING AND POLICING

Per-Stream Filtering and Policing (PSFP) functions allow filtering and policing decisions, and subsequent frame queuing decisions on a per-stream basis.

PSFP is supported by a set of stream filters, stream gates, and flow meters that determine the filtering and policing actions that are to be applied to frames received on the reception port.

PSFP is part of the IEEE 802.1Q, clause 8.6.5, flow classification and metering. IEEE 802.1Q describes PSFP as the last step in the forwarding process before the queuing for output, so that frames that are not forwarded are not policed. Figure 4-12 shows the order of processing.

FIGURE 4-12: PSFP ARCHITECTURE

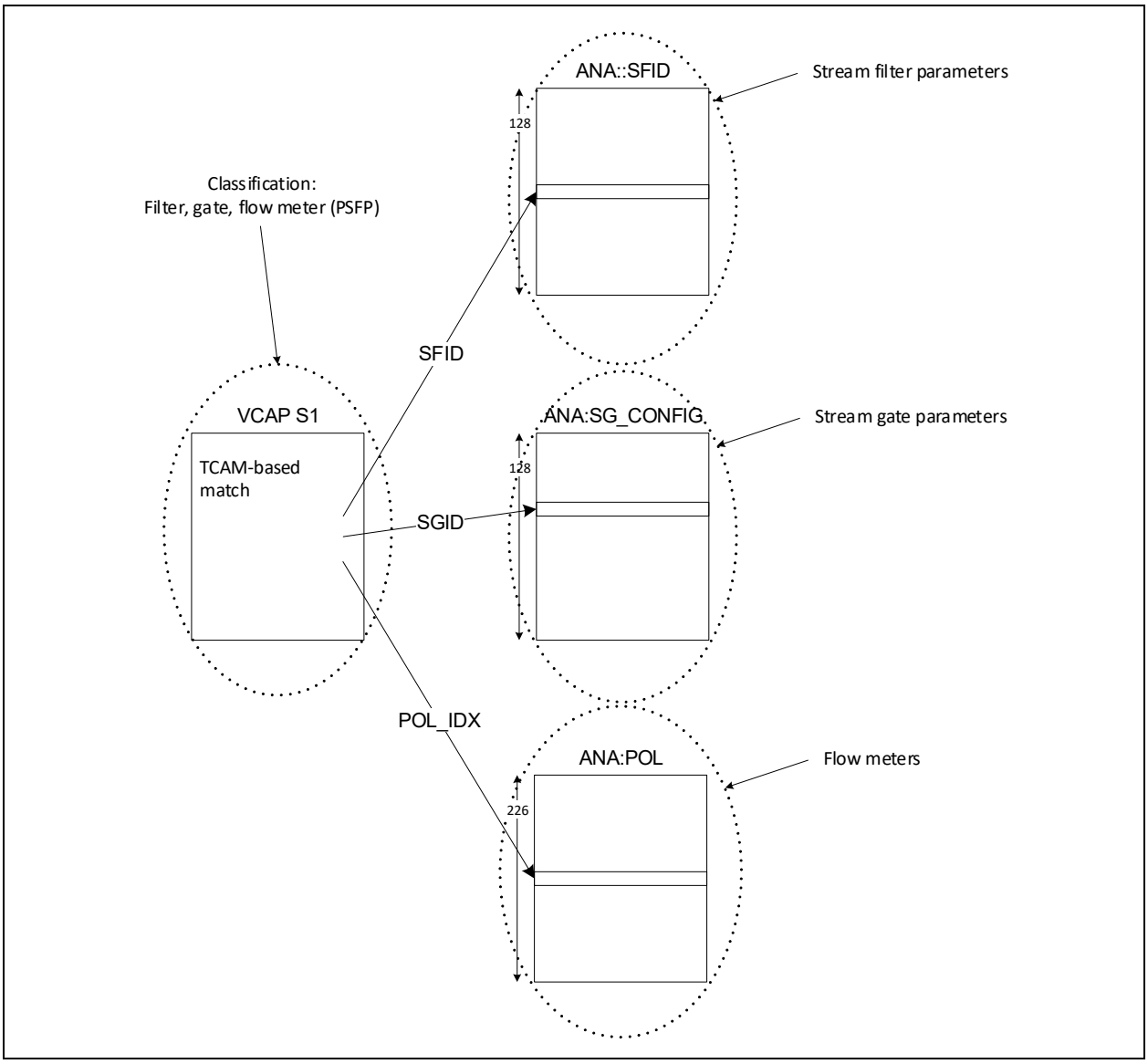


The PSFP support can be described as follows:

- VCAP IS1 classifies stream filters, identified as SFID.
- VCAP IS1 classifies stream gates, identified as SGID.
- VCAP IS1 classifies flow meters, identified as POLICE_IDX.

Figure 4-13 shows the PSFP parameters and associated tables.

FIGURE 4-13: PSFP PARAMETERS AND TABLES



PSFP supports the following functions:

- Max SDU filter: Frames that exceed the maximum SDU size do not pass the stream filter and are dropped.
- Stream gate: The time gates pass or discard frames received according to a repeating schedule that can be synchronized with other systems in the network.
- Flow meter: This refers to a rate policer that enforces bandwidth and burst thresholds.

The stream gates are uniquely identified by the Stream Gate Identifier (SGID) and consist of:

- Operational stream gate state
  - Open: Frames are permitted to pass through the gate.
  - Closed: Frames are not permitted to pass through the gate.
- Internal Priority Value (IPV)
  - Null - Use the existing QoS class derived from the basic classification and VCAP IS1 for queuing.
  - Otherwise - This IPV is used for queuing.

  - Note: The IPV value is only used for queuing. This internal priority value does not replace the QoS class used for frame rewrites in the rewriter.
- Interval octet maximum
  - Specifies the maximum number of MSDU octets permitted in the specified time interval.
  - Default-value of zero indicates that there is no limit on the number of MSDU octets permitted.
  - Otherwise, only configured number of MSDU octets permitted and remaining octets are dropped.
- Stream gate control list
  - Contains an ordered list of stream gate control entries that determines the stream gate state when the control operations are executed based on time.
- Stream gate control state machines
  - Cycle timer state machine
  - List execute state machine
  - List config state machine

For information about the stream gate operation, see Section 4.11.7, Stream Gates.

The device supports a total of 128 global stream filters. These can be distributed across different ports in any possible way.

The device supports a total of 128 stream gate instances. The device supports both many-to-one and one-to-one mapping between SFID entries and SGID entries.

The device supports a total of 226 policer instances. Up to 128 policers can be used for PSFP processing. The device supports both many-to-one and one-to-one mapping between SFID entries and policer instances. For more information on policers, see Section 4.12, Policers.

4.11.6.1 Stream Filter Table

The MAX_SDU filter check is part of the Stream Filter Table. A non-zero value configured to MAX_SDU_LEN will enable this filter. If this filter is enabled, then the frame with size exceeding the configured value is dropped. This check will only work if the frames belonging to this stream is processed in Store/Forward mode. The check is disabled if the frames belonging to this stream are processed in cut through mode of operation.

For BLOCK_OVERSIZE_FRM_ENA feature to have effect, MAX_SDU_LEN should be configured to a non-zero value. The BLOCK_OVERSIZE_FRM_ENA feature also works in cut through mode of operation.

Sometimes, it is necessary to block a faulty stream that is either not coming in its arrival window or is exceeding its bandwidth resources. A stream blocking function is provided for blocking faulty streams (FORCE_BLOCK). If enabled, the stream filter discards all the frames matching this SFID without applying any of the other filters.

The switch management software executing from external CPU should monitor the stream filter statistics, especially the drop counters to check whether any of the drop rates are excessive. In such a case, the faulty stream can be silenced by setting the FORCE_BLOCK bit.

4.11.6.2 Stream Filter Statistics

Table 4-27 list the counters associated with Stream Filter which are maintained by hardware. There are 4 counters which are implemented for each of the SFID.

TABLE 4-27: SFID HARDWARE COUNTERS

Counter	Width	Description
MATCHING_FRAMES_COUNT	32	Counter that counts frames matching the filter and its specifications.
NOT_PASSING_FRAMES_COUNT	32	Counter that counts frames those have not passed the stream gate associated with filter.
NOT_PASSING_SDU_COUNT	32	Counter that counts frames those have not passed the MAX SDU check associated with filter.
RED_FRAMES_COUNT	32	Counter that counts frames those have been dropped by policer pointed to by filter

Table 4-28 lists all the counters which are associated with Per Stream Filtering and Policing. Four counters are implemented in hardware, and two counters must be handled by management software. The two counters that are handled and computed through external management software are indirectly derived from hardware counters.

TABLE 4-28: SFID ALL COUNTERS

Counter	Hardware Implemented	Software Computation	Description
MATCHING_-FRAMES_COUNT	Yes	Read from hardware	Counter that counts frames matching the filter and its specifications.
PASSING_-FRAMES_COUNT	No	MATCHING_FRAMES_COUNT - NOT_PASSING_-FRAMES_COUNT	Counter that counts frames those have passed the stream gate associated with filter.
NOT_PASSING_-FRAMES_COUNT	Yes	Read from hardware	Counter that counts frames those have not passed the stream gate associated with filter.
PASSING_S-DU_COUNT	No	PASSING_FRAMES_COUNT - NOT_PASSING_SDU_COUNT	Counter that counts frames those have passed the MAX SDU check associated with filter.
NOT_PASSING_S-DU_COUNT	Yes	Read from hardware	Counter that counts frames those have not passed the MAX SDU check associated with filter.
RED_-FRAMES_COUNT	Yes	Read from hardware	Counter that counts frames those have been dropped by policer pointed to by filter.

4.11.6.3 PSFP and Cut Through

Cut through forwarding is accomplished by performing two analyses of the frame (see the analysis steps in the forwarding engine as described in Section 4.11.4, Forwarding Engine). The two analyses steps are also referred to as SOF analysis and EOF analysis. For small frames where it is not possible to improve latency in cut through vis-à-vis Store forward mode a combined single SOF+EOF analysis is done.

Like EOF analysis, SOF analysis can do all the analysis steps as described in Section 4.11.4, Forwarding Engine, except the learning and policer updates, which are deferred to EOF analysis. For learning, it is desired that SMAC learning of bad frames are not done and hence it is deferred to EOF analysis, where a frame status (good/bad) is known. For policer updates, it requires knowing the frame size which is only known at the time of EOF analysis.

PSFP can be combined with cut through. For streams which are processed in cut through, PSFP processing is done with SOF analysis. This means that the policer state and stream gate state is checked at the time SOF analysis. If the policer bucket is closed, or the stream gate state is closed, the frame is dropped at SOF analysis. Even if the state changes by the time EOF analysis is done, nothing can be done and the frame is treated to be dropped. This is also true for the opposite condition (i.e., if a policer bucket is open and the stream gate status is open, the frame is accepted at the time of SOF analysis as a Cut Through frame. This decision is not changed if the stream gate closes before EOF analysis is performed). It should be noted that for frames which are directed to a stream gate, the IPV state is used as the QoS value which leads to cut through determination.

For store and forward frames all decisions are made via EOF analysis. It does not matter whether the SOF of the frame came in at a time during which the stream gate was closed.

The device maintains several port statistics and FIFO drop statistics, some of which are based on the QoS value. The IPV value is used as the QoS value for all these statistics. For cut through frames the IPV value at the times of SOF analysis is used. For store and forward frames the IPV value at the time of EOF analysis is used.

4.11.7 STREAM GATES

Stream Gates (SG) are time-based policers used in PSFP. Frames are dropped based on the gate state (OPEN/CLOSE), whose state will be altered based on Gate Control List (GCL) and current PTP time. Apart from time-based policing, SG can alter egress queue selection for the frames that pass through the Gate. This is done through Internal Priority Selector (IPS) as described in IEEE 802.1Q.

The device implements a pool of RAM based stream gates instance tables, which are shared between ports. These stream gates are identified using Stream Gate Identifier (SGID). Further, the hardware supports a total of 128 stream gate instances with a maximum of 4 GCL entries per stream gate instance. Out of these SG instances, valid SG instances are frequently updated for next *CycleStartTime*. This update or refresh rate is 5µs. This brings a limitation on CycleTime, whose value must be minimum 5 µs.

To configure/install the Admin parameters into the Stream Gate (SG) instance, perform the following procedure.

Note: This procedure needs to be repeated for every Stream Gate instance that needs to be installed/re-configured and also make sure that PSFPAdminBaseTime is always set to future time.

1. Configure SGID in ANA::SG_ACCESS_CTRL.SGID for which the Admin parameters need to be installed/re-configured.
2. Enable the SG instance by setting ANA::SG_CONFIG_REG_3.GATE_ENABLE to *1* (this step is not required in the case of re-configuring as the stream gate is already enabled).
3. Configure the rest of the required Admin parameters through ANA::SG_CONFIG_REG_* registers.
4. Configure the GCL entries through ANA::SG_GCL_GS_CONFIG and ANA::SG_GCL_TI_CONFIG registers for the valid GCL entries. TIME_INTERVAL for these entries needs to be cumulative (instead of just time interval as mentioned in IEEE 802.) from zeroth entry to last entry. For example, there are three GCL entries with time intervals t1, t2, and t3, which needs to be installed. t1, t1+t2, and t1+t2+t3 for GCL entry0, entry1, and entry2 respectively are used while configuring ANA::SG_GCL_TI_CONFIG.TIME_INTERVAL. With this, a limitation on CycleTime and sum of TimeInternals exists as CycleTime >= sum of TimeIntervals.
5. Initiate List Config state machine by setting ANA::SG_ACCESS_CTRL.CONFIG_CHANGE to *1*.
6. Wait for the bit, ANA::SG_ACCESS_CTRL.CONFIG_CHANGE, to be cleared.

Note: The timing of the gate operations is linked to the PTP time domain defined in ANA:SG_ACCESS:SG_PTP_DOMAIN_CFG.PTP_DOMAIN. For more information, see Section 4.20, Hardware Time Stamping.

The device provides some Oper domain status values.

The value of PSFPConfigChangeError is always zero as it is assumed that PSFPAdminBaseTime is always set to future time while configuring.

To access these registers for a SG instance (identified using SGID), set ANA::SG_ACCESS_CTRL.SGID to the SGID, and then read the status register.

4.11.8 FRAME REPLICATION AND ELIMINATION FOR RELIABILITY (FRER)

FRER provides increased reliability (reduced packet loss rates) for a stream by:

- Sequencing and replicating every packet, in the source end system and/or in relay systems closer to the source
- Eliminating those replicates in the destination end system and/or in other relay systems closer to the end system.

As far as the device switch capabilities are concerned, the following 802.1CB flows are supported:

- PKT NON_SEQ -> unknown stream -> SEND
- PKT SEQ -> unknown stream -> SEND
- PKT NON_SEQ -> known stream -> SEQ (PUSH R-TAG)-> SPLIT -> SEND(2).
- PKT SEQ -> known stream -> SPLIT -> SEND(2).
- PKT SEQ(2) -> known stream -> DISC -> SEND(1).
- PKT SEQ(2) -> known stream -> DISC (POP R-TAG) -> SEND(1).

Stream identification and sequence generation is implemented at ingress in the analyzer. Individual recovery and sequence recovery functions are implemented at egress in the queue system.

4.11.8.1 Stream Identification

The device supports Null Stream Identification. A stream is identified using (DMAC, VLAN). This is suitable for AVB streams using IEEE 1722 protocol. For a null stream, the encapsulation is not changed when the stream is split or recovered. More advanced passive stream identifications are possible. VCAP IS1 classifies streams, identified as ISDX.

4.11.8.2 Stream Table

The stream table in the device encompasses all the required sequence generation parameters to instantiate FRER in switch on a per stream basis. The stream table is indexed by the stream identifier, ISDX.

The stream table supports 128 streams and each entry in the stream table contains sequence generation parameters.

In addition, the stream table also provides an R-TAG pop parameter used by the rewriter.

Table 4-29 describes the sequence generation parameters.

TABLE 4-29: SEQUENCE GENERATION PARAMETERS

Name	Width	Description
INPUT_PORT_MASK	10	If the packet is from input port belonging to this port mask, then it's a known stream and Sequence generation parameters can be applied
SEQ_GEN_ENA	1	If True, then the Sequence Gen function is instantiated.
STREAM_SPLIT	1	If True, then the Stream splitting function should be instantiated.
GEN_SEQ_NUM	16	The sequence number to be used for outgoing packet passed to SEQ_GEN function.
SPLIT_MASK	9	Port mask used to add redundant paths (or ports) if split is enabled (STREAM_SPLIT) for a stream. This is OR'ed with the final port mask determined by the forwarding engine.
SEQ_GEN_ERR_STATUS	2	Indicates the error status of the stream if the stream is configured for generation. 0: This relay system is expected to sequence the stream, but the incoming packet already has sequence number. 1: The sequence generation is not enabled in this relay system, but the incoming packet is NON-SEQ. In either of the cases the packet causing this condition is forwarded using normal forwarding rules

The sequence generation algorithm as described in the standard is used. However the standard does not describe how to deal with unexpected conditions related to SEQ_GEN_ERR_STATUS as explained above. The following pseudo-code illustrates the sequence generation algorithm.

```
// cfg : Configuration parameters associated with the stream table

frm.seq_type = NON_SEQ / SEQ depends on frame parsing.
REW.seq_op   = REDTAG_NOP;
REW.seq_num  = 0;

// Case of Sequence generation
if (cfg.isdx>0 && !cfg.rtag_pop_ena) {

  // Not a known stream, use normal forwarding
  if (cfg.input_port_mask[frm.igr_port] == 0)
    exit;

  // tsnSidInputPackets variable is incremented once for each frame
  // identified by the Stream Identification
  SidInputPkts[cfg.isdx] ++;

  // Error status(0), use normal forwarding
  if (frm.seq_type == SEQ && cfg.seq_gen_ena)
    Set_error_status(0); exit;

  // Error status(1), use normal forwarding
  if (frm.seq_type == NON_SEQ && !cfg.seq_gen_ena)
    Set_error_status(1); exit;

  if (cfg.seq_gen_ena) {
    REW.seq_num = cfg.gen_seq_num;
    REW.seq_op  = REDTAG_PUSH;
```

```
      if (cfg.gen_seq_num >= (tsnSeqGenSpace - 1))
        cfg.gen_seq_num = 0;
      else
        cfg.gen_seq_num = cfg.gen_seq_num + 1;

      SEND_DATA; // Pass frame down stack with sequence_number subparameter
    }

    if (cfg.stream_split)
      dest_mask |= cfg.split_mask;

    Update_stream_table();
  }
```

Table 4-30 describes the R-TAG pop parameter.

TABLE 4-30: R-TAG POP PARAMETER

Name	Width	Description
RTAG_POP_ENA	1	If True, then the redundancy tag is popped by rewriter.

4.11.8.3 Sequence Generation Statistics

For supporting FRER sequence generation statistics, there is one counter per stream which are stored in hardware. Each counter is 32-bit and can roll over to 0 when incremented past its maximum value.

4.11.8.4 Individual and Sequence Recovery Functions

The individual and sequence recovery functions are located in QSYS. They operate on each egress frame copies made from an ingress frame, after any stream splitting or merging. The device supports 256 individual recovery functions and 128 sequence recovery functions. Both vector and match sequence recovery algorithms are supported. The vector algorithm has a programmable history length with a maximum value of 32. Each egress frame copy is eligible to both an individual recovery function and a sequence recovery function.

Each egress frame copy inputs the following information to the recovery functions:

- ISDX
- Frame's destination port.
- Sequence number.

The FRER_MAP table is indexed with the ISDX and returns a base number for the member stream ID as well as a set of four ports, on which recovery functions can be located. The frame's destination port is matched against the set of ports and a successful match returns the index. The member stream handle is calculated as the base number offset with the index. If the frame's destination port is not matched in the port set returned from FRER_MAP, then no recovery functions are applied to the egress frame copy.

The member stream handle selects which individual recovery function to use. Similarly, the individual recovery function provides the compound stream handle that selects which sequence recovery function to use. It is optional whether the selected recovery functions are enabled. This implies that a given stream can be subject to no recovery at all, only individual recovery, only sequence recovery, or both.

The following pseudo code list the combined behavior of the individual and sequence recovery functions. For simplicity, complexities regarding wrapping of sequence numbers are not considered.

```
// frm : Information provided with the frame
//
// cfg : Configuration parameters associated with the recovery function:
//       EACL:FRER_CFG_MEMBER or EACL:FRER_CFG_COMPOUND
//
// sta : State parameters associated with the recovery function:
//       EACL:FRER_STA_MEMBER or EACL:FRER_STA_COMPOUND
//
// cnt : Counters associated with the recovery function:
//       EACL:FRER_CNT_MEMBER or EACL:FRER_CNT_COMPOUND
```

```
delta = frm.seq_no - sta.sequence_number;
if (cfg.vector_algorithm) {
  out_of_rng = (delta < -cfg.history_length) || (cfg.history_length < delta);
  positive   = (delta > 0);
} else {
  out_of_rng = false;
  positive   = (delta != 0);
}

// Case 1: Frame has no sequence number
if (!frm.seq_no_vld) {
  cnt.cnt_tagless++;
  if (cfg.take_no_sequence) {
    cnt.cnt_passed++;
    if (cfg.vector_algorithm)
      sta.ticks  = cfg.reset_ticks;
  } else {
    cnt.cnt_discarded++;
    drop = true;
  }
// Case 2: Accept all sequence numbers
} else if (take_any) {
  cnt.cnt_passed++;
  sta.sequence_number = frm.seq_no;
  sta.ticks           = cfg.reset_ticks;
  sta.history[0]      = 1; // Add a "seen" bit
  sta.take_any        = false;

// Case 3: Delta is out of range (too far ahead or too far behind)
} else if (out_of_rng) {
  cnt.cnt_rogue++;
  cnt.cnt_discarded++;
  drop = true;
  if (cfg.individual)
    sta.ticks = cfg.reset_ticks;

// Case 4: Delta is positive: Accept new sequence number
} else if (positive) {
  cnt.cnt_passed++;
  if (delta != 0)
    cnt.cnt_outoforder++;

  if (cfg.vector_algorithm) {
    // Increment for each sequence number lost due to shift
    cnt_lost += lost_seq_nums(sta.history, delta);

    // Shift delta "not seen" bits and one "seen bit"
    sta.history = shift(sta.history, delta, 1);
  }
  sta.sequence_number = frm.seq_no;
  sta.ticks           = cfg.reset_ticks;
  sta.take_any        = false;

// Case 5: Sequence number already seen
} else if (!cfg.vector_algorithm || sta.history[-delta] = 1) {
  cnt.cnt_discarded++;
  drop = true;
  if (cfg.individual)
```

```
      sta.ticks = cfg.reset_ticks;

  // Case 6: Accept new sequence number
  } else {
    cnt.cnt_passed++;
    cnt.cnt_outoforder++;

    sta.history[-delta] = 1;
    sta.ticks           = cfg.reset_ticks;
    sta.take_any        = false;
  }
```

Each recovery function uses a watch dog timer that monitors and resets the recovery function when appropriate. All watch dog timers share a common prescaler (FRER_CFG.WATCHDOG_PRESCALER) defining the time base granularity as ticks per second.

Each Individual and Sequence recovery function then configures the number of ticks (FRER_CFG_MEMBER/COMPOUND.RESET_TICKS) that can pass before a reset is triggered. This is the default value as well as the reset value. The current number of ticks left before a reset, is available in FRER_STA_MEMBER/COMPOUND.TICKS.

When resetting a recovery function due to the watch dog timer expiring, the associated reset counter (CNT_MEMBER/COMPOUND_RESETS) is incremented, the sequence number history vector is cleared, the current recovered sequence number is cleared, and the TAKE_ANY state is set to true.

4.11.8.5 Latent Error Detection

The latent error detection functions described in IEEE 802.1CB are executed periodically with periods measured in seconds. Consequentially, these functions are expected to be implemented in software using hardware-based counters.

Each recovery function provides counters for the number of frames passed (CNT_MEMBER/COMPOUND_PASSED.CNT_PASSED) and discarded (CNT_MEMBER/COMPOUND_DISCARDED.CNT_DISCARDED).

4.11.8.6 FRER Individual and Sequence Recovery Statistics

Each recovery function implements a set of 32-bit counters directly corresponding to operational counters defined by IEEE 802.1CB.

Other counters defined by IEEE 802.1CB, for instance frerCpSeqRcvyPassedPackets, must be implemented by means of software by combining hardware-based counters.

4.11.9 CUT-THROUGH

The device supports cut-through forwarding where the switch initiates transmission of a frame before the end of the frame is received. This can lower the frame's latency through the switch, especially for longer frames.

Cut-through is enabled per ingress port per QoS class. In addition, each port's speed must be programmed since a cut-through transfer can only take place to ports of same or lower speed as the source port. This prevents buffer underruns.

Note that cut-through forwarded frames bypass the queue system’s resource management controls including flow control mechanisms. As a result, when cut-through forwarded flows encounter congestion — such as during sustained many-to-one scenarios or when forwarding to ports with lower speeds — this may lead to unexpected frame drops by the queue system. To mitigate the risk of such occurrences, it is essential to exercise caution when enabling cut-through.

4.12 Policers

The device supports 226 policers that can be allocated to ingress ports, QoS classes per port, VCAP IS1 entries, and VCAP IS2 entries. The VCAP IS1 policers are used for PSFP policers while the VCAP IS2 policers are used for access control. The policers limit the bandwidth of received frames by discarding frames exceeding configurable rates. All policers are MEF compliant dual leaky bucket policers that are capable of handling committed and excess peak information rates.

Each frame can hit up to three policers: One port policer, one VCAP IS2 policer, and either a QoS policer or a VCAP IS1 policer. The order in which the policers are applied to a frame is programmable.

In addition to the policers, the device also supports a number of storm policers and an egress scheduler with shaping capabilities. For more information, see Section 4.13.11, Storm Policers and Section 4.14, Scheduler and Shapers.

4.12.1 POLICER ALLOCATION

The different policer types are assigned a policer from the pool of policers the following ways:

- **Port policers:** Frames received on physical port ‘p’ use policer ‘p’. Each of physical ports can be assigned to its own policer.
- **QoS policers:** Frames classified to QoS class ‘q’ on physical port ‘p’ use policer 10 + 8 x ‘p’ + ‘q’. Each of the eight per-port QoS classes can be assigned to its own policer.
- **VCAP IS1 policers:** VCAP IS1 can point to any policer from the pool. If a port policer, a QoS policer, or a VCAP IS1 policer have not allocated a policer from the pool, it is unused and can be assigned as a VCAP IS1 policer. When a frame is assigned a VCAP IS1 policer, then QoS policing is disabled for the frame. The action IS1_ACTION.POLICE_IDX points to the policer that is used.
- **VCAP IS2 policers:** VCAP IS2 can point to any policer from the pool. If a port policer, a QoS policer, or a VCAP IS1 policer have not allocated a policer from the pool, it is unused and can be assigned as a VCAP IS2 policer. The action IS2_ACTION.POLICE_IDX points to the policer that is used.

By default, none of the policers from the pool are allocated.

Port policers are allocated through ANA:PORT:POL_CFG.PORT_POL_ENA per port and QoS policers are allocated through ANA:PORT:POL_CFG.QUEUE_POL_ENA per QoS class per port.

VCAP IS1 policers are allocated by creating IS1 rules with POLICE_ENA and POLICE_IDX actions. IS1 policers actions can be assigned in any of the VCAP S1 lookups.

VCAP IS2 policers are allocated by creating IS2 rules with POLICE_ENA and POLICE_IDX actions. IS2 policers actions are only valid in the first lookup in IS2.

Any frame received by the MAC and forwarded to the classifier is applicable to policing. Frames with errors, pause frames, or MAC control frames are not forwarded by the MAC and, as a result, they are not accounted for in the policers. That is, they are not policed and are not adding to the rate measured by the policers.

In addition, the following special frame types can bypass the policers:

- If ANA:PORT:POL_CFG.POL_CPU_REDIR_8021 is set, frames being redirected to the CPU due to the classifier detecting the frames as being BPDU, ALLBRIDGE, GARP, or CCM/Link trace frames are not policed.
- If ANA:PORT:POL_CFG.POL_CPU_REDIR_IP is set, frames being redirected to the CPU due to the classifier detecting the frames as being IGMP or MLD frames are not policed.

These frames are still considered part of the rates being measured so the frames add to the relevant policer buckets but they are never discarded due to policing.

The VCAP IS2 has the option to disable all policing except the VCAP IS2 policer. This is done with the action POLICE_VCAP_ONLY. If POLICE_VCAP_ONLY is set, only a VCAP assigned policer can police the frame. The other policers are inactive meaning the frame does not add to the policers’ buckets and the frame is never discarded due to policing by the policers.

The order in which the policers are executed is controlled through ANA:PORT:POL_CFG.POL_ORDER. The order can take the following main modes:

- **Serial** The policers are checked one after another. If a policer is closed, the frame is discarded and the subsequent policer buckets are not updated with the frame. The serial order is programmable.
- **Parallel with independent bucket updates** The three policers are working in parallel independently of each other. Each frame is added to a policer bucket if the policer is open, otherwise the frame is discarded. A frame may be added to one policer although another policer is closed.
- **Parallel with dependent bucket updates** The three policers are working in parallel but dependent on each other with respect to bucket updates. A frame is only added to the policer buckets if all three policers are open.

4.12.2 POLICER BURST AND RATE CONFIGURATION

Each of the policers is MEF-compliant dual leaky bucket policers. This implies that each policer supports the following configurations:

- **Committed Information Rate (CIR):** Specified in POL_CIR_CFG.CIR_RATE in steps of 33.3 kbps. Maximum rate is 1.09 Gbps
- **Committed Burst Size (CBS):** Specified in POL_CIR_CFG.CIR_BURST in steps of 4 kilobytes. Maximum is 240 kilobytes.
- **Excess Information Rate (EIR):** Specified in POL_PIR_CFG.PIR_RATE in steps of 33.3 kbps. Maximum rate is 1.09 Gbps.

- **Excess Burst Size (EBS):** Specified in POL_PIR_CFG.PIR_BURST in steps of 4 kilobytes. Maximum is 240 kilobytes.
- **Coupling flag:** If POL_MODE_CFG.DLB_COUPLED is set, frames classified as yellow (DP level = 1) are allowed to use of the committed information rate when not fully used by frames classified as green (DP level = 0). If cleared, the rate of frames classified as yellow are bounded by EIR.
- **Color mode:** Color-blind or color-aware. A policer always obey the frame color assigned by the classifier. To achieve color-blindness, the classifier must be set up to classify all incoming frames to DP level = 0.

The following parameters can also be configured per policer:

- The leaky bucket calculation can be configured to include or exclude preamble and inter-frame gap through configuration of POL_MODE_CFG.IPG_SIZE.
- Each policer can be configured to measure frame rates instead of bit rates (POL_MODE_CFG.FRM_MODE). The rate unit can be configured to 100 frames per second or 1 frame per second.
- Each policer can operate as a single leaky bucket by disabling POL_MODE_CFG.CIR_ENA. When operating as a single leaky bucket, the POL_PIR_CFG register controls the rate and burst of the policer.

By default, a policer discards frames while the policer is closed. A discarded frame is neither forwarded to any ports (including the CPU) nor is it learned.

Each port policer, however, has the option to run in flow control where the policer instructs the MAC to issue flow control pause frames instead of discarding frames. This is enabled in ANA:PORT:POL_FLOWC. Common for all port policers, POL_HYST.POL_FC_HYST specifies a hysteresis, which controls when the policer can re-open after having closed.

To improve fairness between small and large frames being policed by the same policer, POL_HYST.POL_DROP_HYST specifies a hysteresis that controls when the policer can re-open after being closed. By setting it to a value larger than the maximum transmission unit, it guarantees that when the policer opens again, all frames have the same chance of being accepted. This setting only applies to policers working in drop mode.

The current fill level of the dual leaky buckets can be read in POL_PIR_STATE and POL_CIR_STATE. The unit is 0.5 bits.

4.12.3 PSFP FLOW METERS

In addition to the MEF 10.3 functionality, the policers support the following PSFP specific features.

- **DropOnYellow:** A policer is enabled for DropOnYellow=TRUE using ANA:POL:POL_-MODE_CFG.DROP_ON_YELLOW_ENA.
- **MarkAllFramesRed/MarkAllFramesRedEnable:** A policer is enabled for MarkAllFramesRed using ANA:POL:POL_MODE_CFG.MARK_ALL_FRMS_RED_ENA.

To get notifications about a policer having entered MarkAllFramesRed=TRUE, SW must poll ANA:POL:POL_STATE.MARK_ALL_FRMS_RED_SET for all policers with MARK_ALL_FRMS_RED_ENA set. To change a policer from state MarkAllFramesRed=TRUE to MarkAllFramesRed=FALSE, SW must clear the above notification.

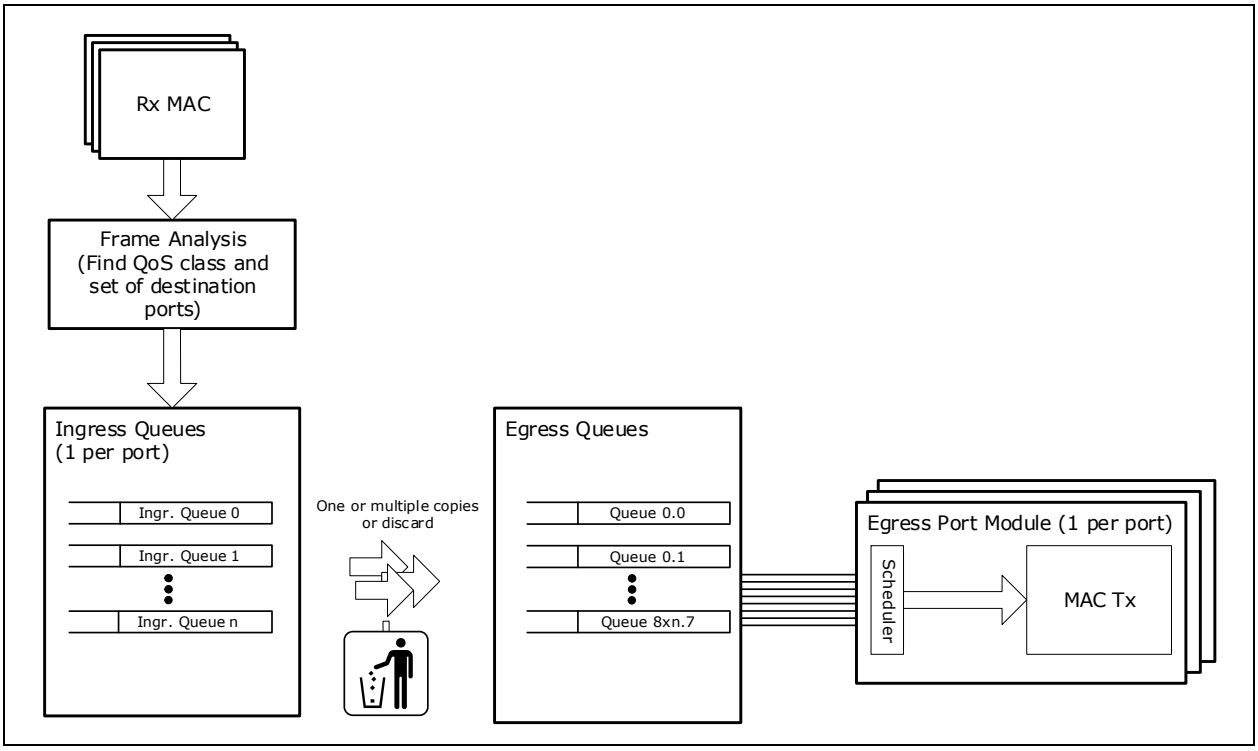
4.13 Shared Queue System

The device includes a shared queue system with one ingress queue per ingress port and one egress queue per QoS class per egress port per ingress port. The queue system has 160 kilobytes of buffer.

Frames are linked into the ingress queues after frame analysis. Each egress port module selected by the frame analysis receives a link to the frame and stores the link in the appropriate egress queue selected by mapping various frame properties to a queue number. Each egress port module has a scheduler that selects between the egress queues when transmitting frames.

Figure 4-14 shows the shared queue system.

FIGURE 4-14: QUEUE SYSTEM OVERVIEW



Resource depletion can prevent one or more of the frame copies from the ingress queue to the egress queues. If a frame copy cannot be made due to lack of resources, the ingress port's flow control mode determines the behavior as follows:

- Ingress port is in drop mode: The frame copy is discarded.
- Ingress port is in flow control mode: The frame is held back in the ingress queue and the frame copy is made when the congestion clears.

For more information about special configurations of the shared queue system with respect to flow control, see Section 4.13.7, Ingress Pause Request Generation.

4.13.1 BUFFER MANAGEMENT

A number of watermarks control how much data can be pending in the egress queues before the resources are depleted. There are no watermarks for the ingress queues, except for flow control, because the ingress queues are empty most of the time due to the fast transfer rates from ingress to egress. For more information, see Section 4.13.7, Ingress Pause Request Generation. When the watermarks are configured properly, congested traffic does not influence the forwarding of non-congested traffic.

The memory is split into two main areas:

- A reserved memory area. The reserved memory area is subdivided into areas per port per QoS class per direction (ingress/egress).
- A shared memory area, which is shared by all traffic.

For setting up the reserved areas, egress watermarks exist per port and per QoS class for both ingress and egress.

All the watermarks, including the ingress watermarks, are compared against the memory consumptions in the egress queues. For example, the ingress watermarks in BUF_Q_RSRV_I compare against the total consumption of frames across all egress queues received on the specific ingress port and classified to the specific QoS class. The ingress watermarks in BUF_P_RSRV_I compare against the total consumption of all frames across all egress queues received on the specific ingress port.

The reserved areas are guaranteed minimum areas. A frame cannot be discarded or held back in the ingress queues if the frame's reserved areas are not yet used.

The shared memory area is the area left when all the reservations are taken out. The shared memory area is shared between all ports, however, it is possible to configure a set of watermarks per QoS class and per drop precedence level (green/yellow) to stop some traffic flows before others.

The sharing watermarks are maximum areas in the shared memory that a given traffic flow can use. They do not guarantee anything.

When a frame is enqueued into the egress queue system, the frame first consumes from the queue's reserved memory area, then from the port's reserved memory area. When all the frame's reserved memory areas are full, it consumes from the shared memory area.

The following provides some simple examples on how to configure the watermarks and how that influences the resource management.

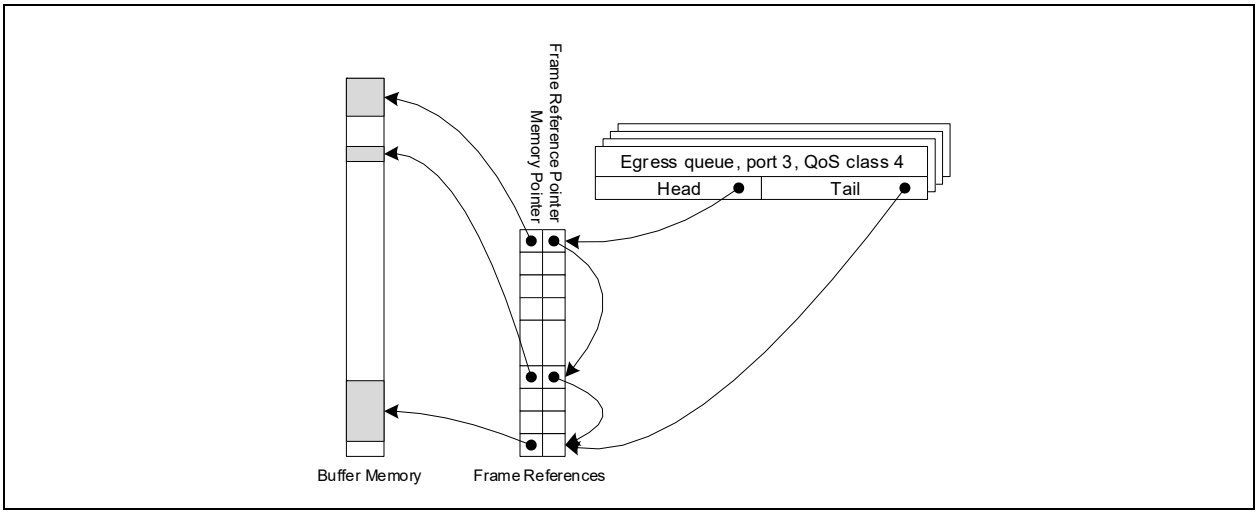
- Setting BUF_Q_RSRV_E(egress port = 6, QoS class = 4) to 2 kilobytes guarantees that traffic destined for port 6 classified to QoS class 4 have room for 2 kilobytes of frame data before frames can get discarded.
- Setting BUF_Q_RSRV_I(ingress port = 6, QoS class = 4) to 2 kilobytes guarantees that traffic received on port 6 classified to QoS class 4 have room for 2 kilobytes of frame data before frames can get discarded.
- Setting BUF_P_RSRV_I(ingress port 6) to 10 kilobytes guarantees that traffic received on port 6 have room for 10 kilobytes of data before frames can get discarded.
- The three above reservations reserve in total 14 kilobytes of memory (2 + 2+ 10 kilobytes) for port 6. If the same reservations are made for all ports, there are 160 – (7 × 14) = 62 kilobytes left for sharing. If the sharing watermarks are all set to 62 kilobytes, all traffic groups can consume memory from the shared memory area without restrictions.

  If, instead, setting BUF_PRIO_SHR_E(QoS class = 7) to 62 kilobytes and the other watermarks BUF_PRIO_SHR_E(QoS class = 0:6) to 14 kilobytes guarantees that traffic classified to QoS class 7 has 48 kilobytes extra buffer. The buffer is shared between all ports.
- The BUF_PRIO_SHR_I, BUF_PRIO_SHR_E, REF_PRIO_SHR_I, and REF_PRIO_SHR_E watermarks can be used for guaranteeing shared resources for each individual QoS class. This is done by setting QSYS::RES_QOS_MODE so that the watermarks operate on the current consumption of each QoS class instead of the total use of the shared resources.

4.13.2 FRAME REFERENCE MANAGEMENT

Each frame in an egress queue consumes a frame reference, which is a pointer element that points to the frame's data in the memory and to the pointer element belonging to the next frame in the queue. Figure 4-15 shows how the frame references are used for creating the queue structure.

FIGURE 4-15: FRAME REFERENCE



The shared queue system holds a table of 1280 frame references. The consumption of frame references is controlled through a set of watermarks. The set of watermarks is the exact same as for the buffer control. The frame reference watermarks are prefixed REF_. Instead of controlling the amount of consumed memory, they control the number of frame references. Both reservation and sharing watermarks are available.

When a frame is enqueued into the shared queue system, the frame consumes first from the queue's reserved frame reference area, then from the port's reserved frame reference area. When all the frame's reserved frame reference areas are full, it consumes from the shared frame reference area.

4.13.3 RESOURCE DEPLETION CONDITION

A frame copy is made from an ingress port to an egress port when both a memory check and a frame reference check succeed. The memory check succeeds when at least one of the following conditions is met.

- Ingress memory is available: BUF_Q_RSRV_I or BUF_P_RSRV_I are not exceeded.
- Egress memory is available: BUF_Q_RSRV_E or BUF_P_RSRV_E are not exceeded.
- Shared memory is available: None of BUF_PRIO_SHR_E, BUF_COL_SHR_E, BUF_PRIO_SHR_I, or BUF_COL_SHR_I are exceeded.

The frame reference check succeeds when at least one of the following conditions is met.

- Ingress frame references are available: REF_Q_RSRV_I or REF_P_RSRV_I are not exceeded.
- Egress frame references are available: REF_Q_RSRV_E or REF_P_RSRV_E are not exceeded.
- Shared frame references are available: None of REF_PRIO_SHR_E, REF_COL_SHR_E, REF_PRIO_SHR_I, or REF_COL_SHR_I are exceeded.

4.13.4 CONFIGURATION EXAMPLE

This section provides an example of how the watermarks can be configured for a QoS-aware switch with no color handling and the effects of the settings.

TABLE 4-31: WATERMARK CONFIGURATION EXAMPLE

Watermark	Value	Comment
BUF_Q_RSRV_I	500 bytes	Guarantees that a port is capable of receiving at least one frame in all QoS classes. **Note:** It is not necessary to assign a full MTU, because the watermarks are checked before the frame is added to the memory consumption.
BUF_P_RSRV_I	0	No additional guarantees for the ingress port.
BUF_Q_RSRV_E	200 bytes	Guarantees that all QoS classes are capable of sending a non-congested stream of traffic through the switch.
BUF_P_RSRV_E	10 kilobytes	Guarantees that all egress ports have 10 kilobytes of buffer, independently of other traffic in the switch. This is the most demanding reservation in this setup, reserving 70 kilobytes of the total 160 kilobytes.
BUF_COL_SHR_E BUF_COL_SHR_I	Maximum	Effectively disables frame coloring as watermark is never reached.
BUF_PRIO_SHR_E BUF_PRIO_SHR_I	4 kilobytes to 32 kilobytes	The different QoS classes are cut-off with 3 kilobytes distance (11, 24, 27, 20, 23, 26, 29, and 32 kilobytes). This gives frames with higher QoS classes a larger part of the shared buffer area. Effectively, this means that the burst capacity is 24 kilobytes for frames belonging to QoS class 0 and up to 42 kilobytes for frame belonging to QoS class 7.
REF_Q_RSRV_E REF_Q_RSRV_I	4	For both ingress and egress, this guarantees that four frames can be pending from and to each port.
REF_P_RSRV_E REF_P_RSRV_I	20	For both ingress and egress, this guarantees that an extra 20 frames can be pending, shared between all QoS classes within the port.
REF_COL_SHR_E REF_COL_SHR_I	Maximum	Effectively disables frame coloring as watermark is never reached.

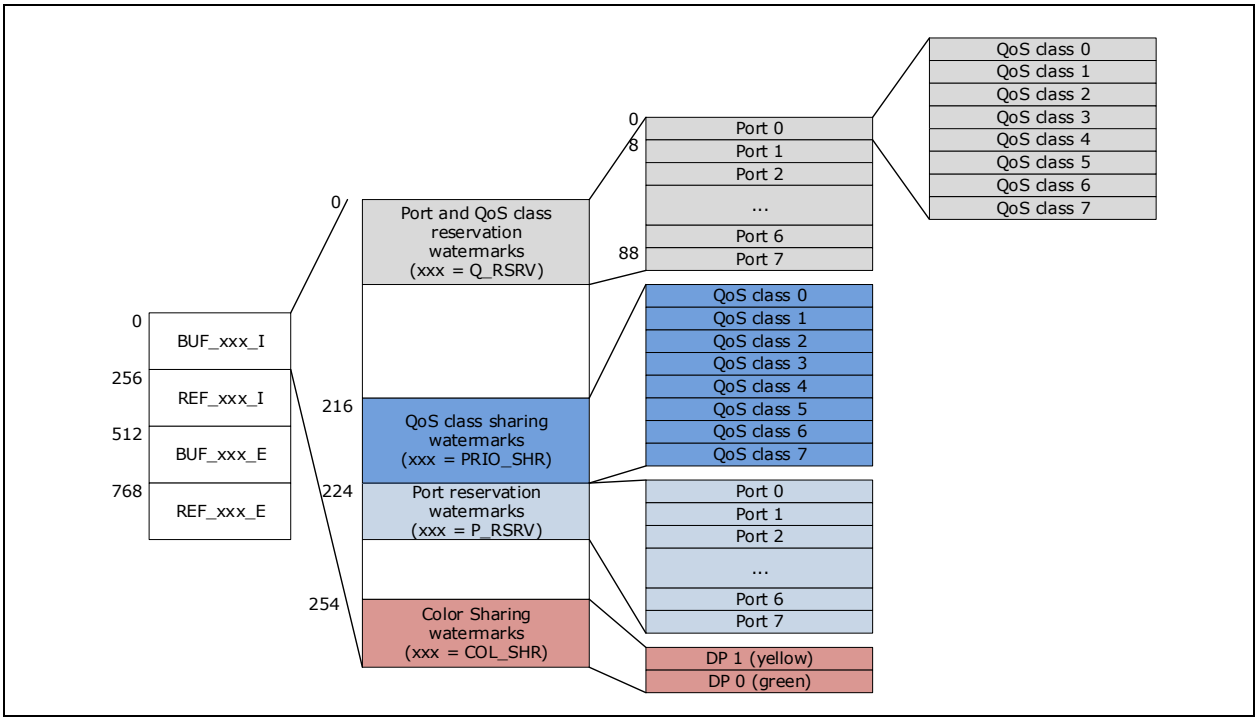
TABLE 4-31: WATERMARK CONFIGURATION EXAMPLE (CONTINUED)

Watermark	Value	Comment
REF_PRIO_SHR_E REF_PRIO_SHR_I	350 - 700	The different QoS classes are cut-off with a distance of 50 frame references (350, 400, 450, 500, 550, 600, 650, and 700). This gives frames with higher QoS classes a larger part of the shared reference area.

4.13.5 WATERMARK PROGRAMMING AND CONSUMPTION MONITORING

The watermarks previously described are all found in the SYS::RES_CFG register. The register is replicated 1024 times. Figure 4-16 shows the organization.

FIGURE 4-16: WATERMARK LAYOUT



The illustration shows the watermarks available for the BUF_xxx_I group of watermarks. For the other groups of watermarks (BUF_xxx_I, REF_xxx_I, BUF_xxx_E, and REF_xxx_E), the exact same set of watermarks are available.

For monitoring the consumption of resources, SYS::RES_STAT provides information about current use and the maximum use since the last read of the register. The information is available for each of the watermarks listed and the layout of the RES_STAT register follows the layout of the watermarks. SYS::MMGT.FREECNT holds the amount of free memory in the shared queue system and SYS::EQ_CTRL.FP_FREE_CNT holds the number of free frame references in the shared queue system.

4.13.6 ADVANCED RESOURCE MANAGEMENT

A number of additional handles into the resource management system are available for special use of the device. They are described in Table 4-32.

TABLE 4-32: RESOURCE MANAGEMENT

Resource Management	Description
Forced drop of egress and ingress frames	QSYS::EGR_DROP_MODE QSYS::SWITCH_PORT_MODE.INGRESS_DROP_MODE. If either an ingress port or an egress port in a frame transfer are configured for drop mode, congestion results in frame discards. Otherwise frames are held back in the ingress queues with potential head-of-line blocking effects. Normally all egress ports are set to non-drop-mode while the ingress drop mode reflects whether or not the port is configured for flow control.
Prevent ingress port from using of the shared resources.	QSYS::IGR_NO_SHARING. For frames received on ports set in this mask, the shared watermarks are considered exceeded. This prevents the port from using more resources than allowed by the reservation watermarks.
Prevent egress port from using of the shared resources.	QSYS::EGR_NO_SHARING. For frames switched to ports set in this mask the shared watermarks are considered exceeded. This prevents the port from using more resources than allowed by the reservation watermarks.
Weighted Random Early Detection (WRED)	QSYS::RED_PROFILE. It is possible to discard frames with increasing probability as the consumption of shared resources per QoS class per drop precedence level increases. QSYS::RED_PROFILE configures a low and a high watermark per QoS class per drop precedence level. The probability of discarding a frame increases linearly from 0% when the consumption is at the low watermark to 100% when the consumption exceeds the high watermark.
Prevent dequeuing	QSYS::PORT_MODE.DEQUEUE_DIS. Each egress port can disable dequeuing of frames from the egress queues.

4.13.7 INGRESS PAUSE REQUEST GENERATION

During resource depletion, the shared queue system either discards frames when the ingress port operates in drop mode, or holds back frames when the ingress port operates in flow control mode. The following describes special configuration for the flow control mode.

The shared queue system is enabled for holding back frames during resource depletion in SYS:PORT:PAUSE_CFG.PAUSE_ENA. In addition, this enables the generation of pause requests to the port module based on memory consumptions. The MAC uses the pause request to generate pause frames or create back pressure collisions to halt the link partner. This is done according to the MAC configuration. For more information about MAC configuration, see Section 4.7.1, Media Access Controller (MAC).

The shared queue system generates the pause request based on the ingress port's memory consumption and also based on the total memory consumption in the shared queue system. This enables a larger burst capacity for a port operating in flow control while not jeopardizing the non-dropping flow control.

Generating the pause request partially depends on a memory consumption flag, TOT_PAUSE, which is set and cleared under the following conditions:

- The TOT_PAUSE flag is set when the total consumed memory in the shared queue system exceeds the SYS:PORT:PAUSE_TOT_CFG.PAUSE_TOT_START watermark.
- The TOT_PAUSE flag is cleared when the total consumed memory in the shared queue system is below the SYS:PORT:PAUSE_TOT_CFG.PAUSE_TOT_STOP watermark.

The pause request is asserted when both of the following conditions are met.

- The TOT_PAUSE flag is set.
- The ingress port memory consumption exceeds the SYS:PORT:PAUSE_CFG.PAUSE_START watermark.

The pause request is deasserted the following condition is met:

- The ingress port's consumption is below the SYS:PORT:PAUSE_CFG.PAUSE_STOP watermark.

4.13.8 TAIL DROPPING

The shared queue system implements a tail dropping mechanism where incoming frames are discarded if the port's memory consumption and the total memory consumption exceed certain watermarks. Tail dropping implies that the frame is discarded unconditionally. All ports in the device are subject to tail dropping. It is independent of whether the port is in flow control mode or in drop mode.

Tail dropping can be effective under special conditions. For example, tail dropping can prevent an ingress port from consuming all the shared memory when pause frames are lost or when the link partner is not responding to pause frames.

The shared queue system initiates tail dropping by discarding the incoming frame if the following two conditions are met at any point while writing the frame data to the memory.

- If the Ingress port memory consumption exceeds the SYS:PORT:ATOP_CFG.ATOP watermark
- If the total consumed memory in the shared queue system exceeds the SYS:PORT:ATOP_TOT_CFG.ATOP_TOT watermark

4.13.9 TEST UTILITIES

Test utilities are built into the shared queue system.

Each egress port can enable a frame repeater (SYS::REPEATER), which means that the head-of-line frames in the egress queues are transmitted but not dequeued after transmission. As a result, the scheduler sees the same frames again and again while the repeater function is active.

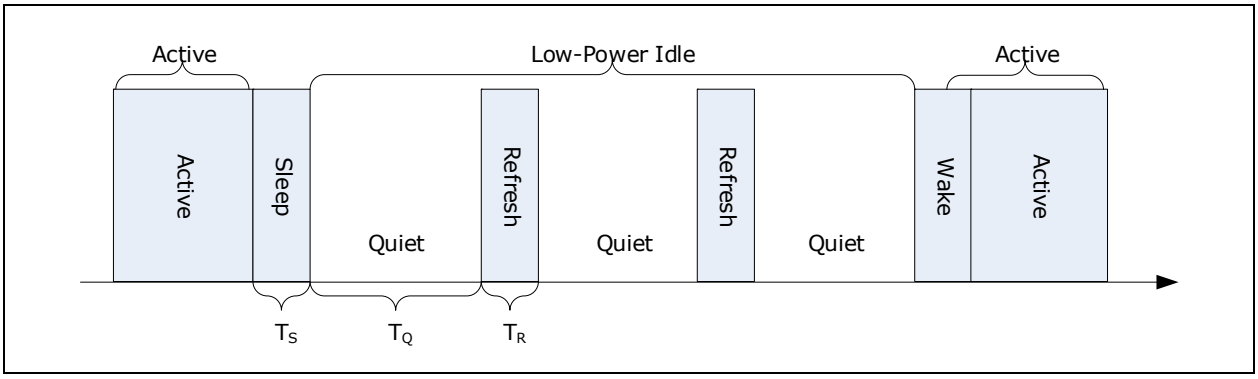
The SYS:PORT:PORT_MODE.DEQUEUE_DIS disables both transmission and dequeuing from the egress queues when set.

4.13.10 ENERGY EFFICIENT ETHERNET

This section provides information about the functions of Energy Efficient Ethernet in the shared queue system.

The shared queue system supports Energy Efficient Ethernet (EEE) as defined by IEEE Draft P802.3az by initiating the Low Power Idle (LPI) mode during periods of low link use. EEE is controlled per port by an egress queue state machine that monitors the queue fillings and ensures correct wake-up and sleep timing. The egress queue state machine is responsible for informing the PCS in the port module of changes in EEE states (active, sleep, low power idle, and wake up).

FIGURE 4-17: LOW POWER IDLE OPERATION



Energy Efficient Ethernet is enabled per port through `DEV::EEE_CFG.EEE_ENA`.

By default, the egress port is transmitting enqueued data. This is the active state. If none of the port's egress queues have enqueued data for the time specified in DEV::EEE_CFG.EEE_TIMER_HOLDOFF, the egress port instructs the PCS to enter the EEE sleep state.

When data is enqueued in any of the port's egress queues, a timer (DEV::EEE_CFG.EEE_TIMER_AGE) is started. If one of the following conditions is met, the port enters the wake up state.

- A queue specified as high priority (QSYS:PORT:EEE_CFG.EEE_FAST_QUEUES) has any data to transmit.

- The total number of frames in the port's egress queues exceeds QSYS::EEE_THRES.EEE_HIGH_FRAMES.
- The total number of bytes in the port's egress queues exceeds QSYS::EEE_THRES.EEE_HIGH_FRAMES.
- The time specified in DEV::EEE_CFG.EEE_TIMER_AGE has passed. PCS is instructed to wake up.

To ensure that PCS, PHY, and link partner are resynchronized after waking up; the egress port holds back transmission of data until the time specified in DEV::EEE_CFG.EEE_TIMER_WAKEUP has passed. After this time interval, the port resumes transmission of data.

The status bit DEV::EEE_CFG.PORT_LPI is set while the egress port holds back data due to LPI (from the sleep state to the wake up state, both included).

4.13.11 STORM POLICERS

The storm policers are activated during frame forwarding.

The queue system contains three storm policers that can limit the maximum allowed forwarding frame rate for broadcast, multicast, and unicast traffic. The storm policers are common to all ports and, as a result, measure the sum of traffic forwarded by the switch. A frame can activate only one of the storm policers, and the frame is discarded if the activated storm policer exceeds the configured rate.

Each storm policer can be configured to a frame rate ranging from 1 frame per second to 1 million frames per second.

The traffic types eligible for storm policing are configurable through ANA::STORM_CFG. By default, frames redirected to the CPU bypass the policers. This behavior can be changed in STORM_CFG.STORM_NO_POL_DIS.

The storm policers differentiate between frames that are flooded and frames that are forwarded to a known destination. This differentiation is the result of whether the analyzer finds a match in the PMAC table or the MAC table for the (VID, DMAC) pair. See Section 4.11.4.1, DMAC Analysis for more details on the PMAC and MAC table lookups.

For the broadcast storm policer, the following traffic types can be included (STORM_CFG.STORM_BC_MASK):

- Flooded frames with DMAC = 0xFFFFFFFFFFFF.
- Forwarded frames with DMAC = 0xFFFFFFFFFFFF.

For the multicast storm policer, the following traffic types can be included (STORM_CFG.STORM_MC_MASK and STORM_CFG.STORM_IPMC_MASK):

- Flooded IP multicast frames with DMAC = 0x01005E000000 to 0x01005E7FFFFF
- Forwarded IP multicast frames with DMAC = 0x01005E000000 to 0x01005E7FFFFF
- Flooded frames with DMAC bit 40 set, except IP multicast and broadcast
- Forwarded frames with DMAC bit 40 set, except IP multicast and broadcast

For the unicast storm policer, the following traffic types can be included (STORM_CFG.STORM_BC_MASK):

- Flooded frames with DMAC bit 40 cleared
- Forwarded frames with DMAC bit 40 cleared

For each of the storm policers, a maximum rate is configured in STORMLIMIT_CFG and STORMLIMIT_BURST:

- STORM_UNIT chooses between a base unit of 1 frame per second or 1 kiloframes per second.
- STORM_RATE sets the rate to 1, 2, 4, 8, …, 1024 times the base unit (STORM_UNIT).
- STORM_BURST configures the maximum number of frames in a burst.
- STORM_MODE specifies how the storm policer affects the forwarding decision. The options are:
  - Disabled. Policing has no effect on the forwarding of the frame.
  - When policing, clear CPU extraction queues only.
  - When policing, clear front port destination only.
  - When policing, clear both CPU extraction queues and front port destinations.

Note that if an NPI port is used for one or more CPU extraction queues, then STORM_MODE treats the NPI port as CPU extraction queues and not as a front port.

Each policer keeps a count of frames discarded by the policer (STORMLIMIT_STAT).

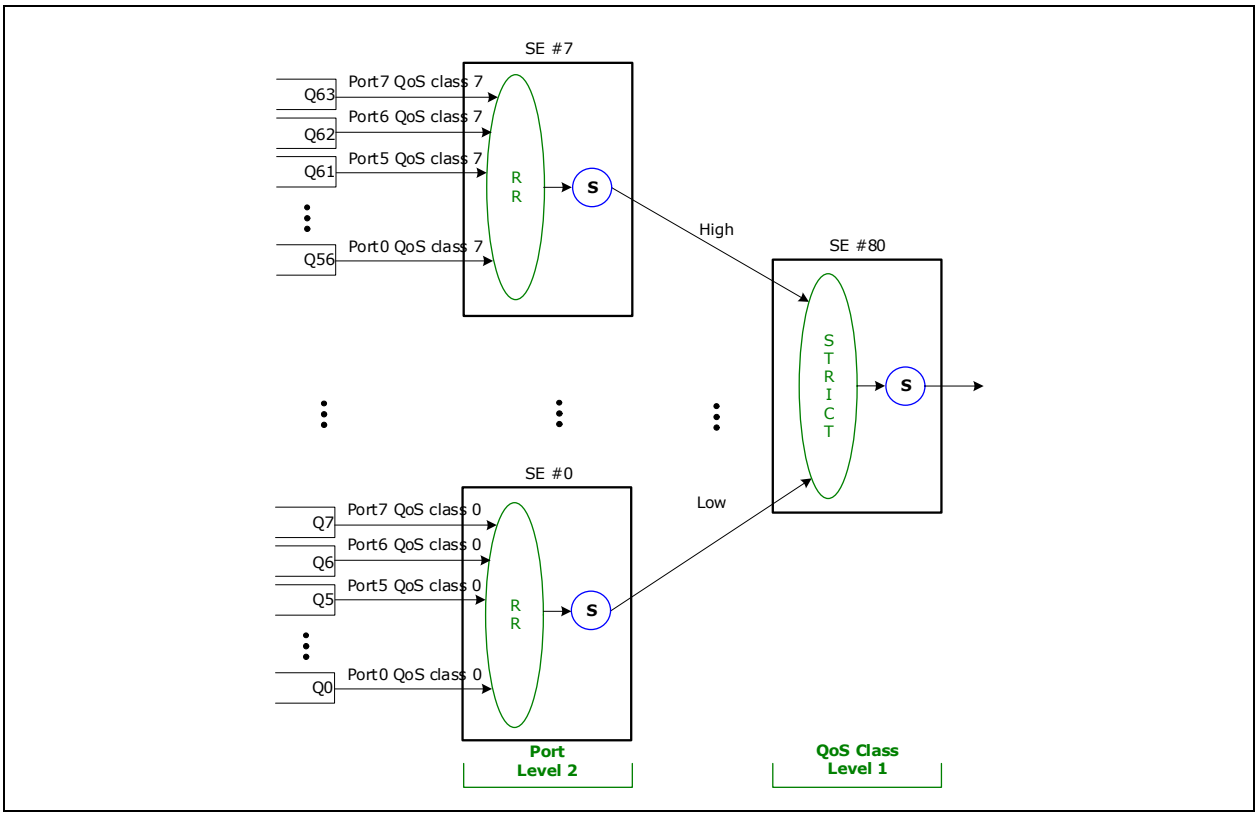
4.14 Scheduler and Shapers

Each egress port contains a two-level priority-fair egress scheduler. The first scheduler level towards the egress port schedules between QoS classes while the second scheduler level towards the egress queues schedules between the ingress ports.

An egress scheduler is constructed using 9 scheduler elements. Each scheduler elements has 8 inputs and 1 output. It contains a scheduler, which can be strict or mixed with a round robin based scheduling algorithm. The round robin based scheduling algorithm can either be frame-based round robin or byte-based deficit weighted round robin. The output port of a scheduler elements contains a dual leaky bucket shaper.

Figure 4-18 provides an overview of the egress scheduling system for egress port 0.

FIGURE 4-18: EGRESS SCHEDULER PORT 0



Each egress port features a similar egress scheduler. Table 4-33 lists which scheduler elements are used by the different ports.

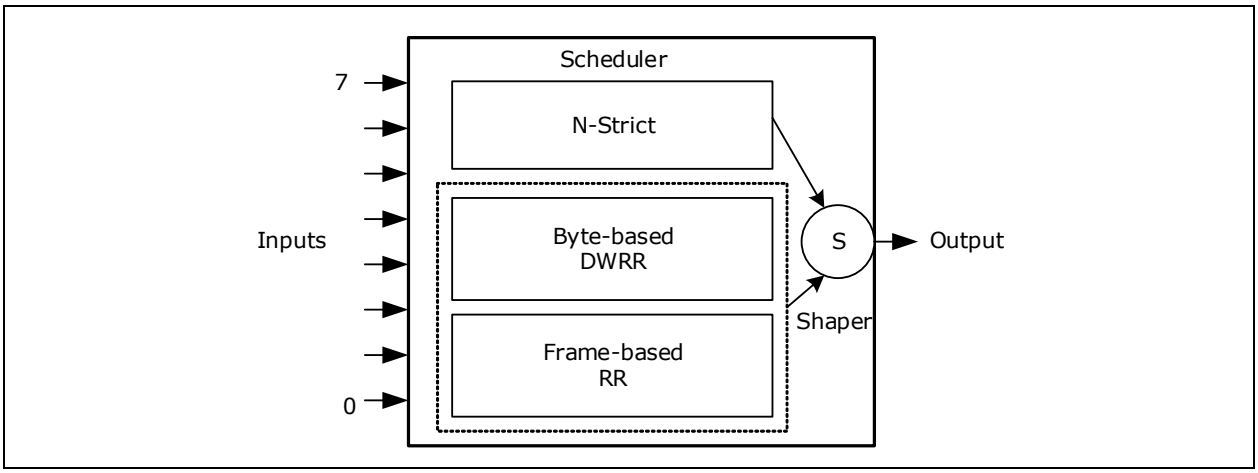
TABLE 4-33: SCHEDULER ELEMENTS NUMBERING

Egress Port	Scheduler Elements - Level 1	Scheduler Elements - Level 2	Queues
0	80	0 through 7	0 through 7
1	81	8 through 15	8 through 15
2	82	16 through 23	16 through 23
3	83	24 through 31	24 through 31
4	84	32 through 39	32 through 39
5	85	40 through 47	40 through 47
6	86	48 through 55	48 through 55
7	87	56 through 63	56 through 63
8	88	64 through 71	64 through 71
9 (CPU)	89	72 through 79	72 through 79
10 (CPU)	90	80 through 87	80 through 87

4.14.1 SCHEDULER ELEMENT

Figure 4-19 shows the scheduler element.

FIGURE 4-19: SCHEDULER ELEMENT



By default, the scheduler operates in strict priority between all 8 inputs. The inputs are searched in the following prioritized order: Input 7 has highest priority followed by 6, 5, 4, 3, 2, 1, and 0.

In addition, the scheduler can operate either in a byte-based deficit weighted round robin (DWRR) mode or a frame-based round robin (RR) mode. This is an overall configuration for the scheduler element and cannot be selected per input. In DWRR mode, the participating inputs are given a weight and the scheduler selects frames from these inputs according to the weights. The DWRR is byte-based and takes the lengths of the frames into account. In RR mode, the participating inputs are selected one after another. The RR is frame-based and does not take the length of the frames into account.

The scheduler supports a mixed mode where some inputs operate in strict priority and others operate in either DWRR or RR. Any number of inputs can be assigned to either group but strict priority inputs must always be selected from highest numbered inputs.

Each scheduler element has an associated leaky-bucket shaper at the output. The shaper limits the overall transmission bandwidth from the scheduler element. Frames are only scheduled if the shaper is open.

Each scheduling element determines whether it has a frame ready for scheduling based on status information of the scheduling element's inputs. For each input, the scheduling element knows if the input has a frame ready for transmission and if the frame is ready due to a work conservation mode. The overall scheduling algorithm within a scheduling element is as follows:

1. If the output shaper is closed, frames are only scheduled from the element if the element is enabled for work conservation. Otherwise, frames are held back until the shaper opens.
2. If the output shaper is open or in work conservation mode, the scheduling element schedules between inputs that are not work conserving by following the rules for the scheduler configuration: strict inputs are scheduled first then round robin based inputs are scheduled according to either the DWRR algorithm or the RR algorithm.
3. If no frames are scheduled during step 2, a second round of scheduling is performed. Inputs that are work conserving become candidates for the second round of scheduling.

The hierarchy of scheduling elements is traversed from the element connected to the egress port through to the element that connects to an egress queue by recursively deciding which input should be scheduled.

4.14.2 EGRESS SHAPERS

Each of the scheduling elements contain an output shaper with the following configurations:

- Committed maximum rate – Specified in CIR_CFG.RATE in steps of 100 kbps. Maximum is 3.28 Gbps.
- Committed maximum burst size – Specified in CIR_CFG.BURST in steps of 4 kilobytes. Maximum is 252 kilobytes.

The shaper can operate byte-based or frame-based (SE_CFG.SE_FRM_MODE). When operating as byte-based, the frame adjustment value HSCH_MISC.FRM_ADJ can be used to program the fixed number of extra bytes to add to each frame transmitted (irrespective of QoS class) in the shaper and DWRR calculations. A value of 20 bytes corresponds to line-rate calculation and accommodates for 12 bytes of inter-frame gap and 8 bytes of preamble. Data-rate based shaping and DWRR calculations are achieved by programming 0 bytes.

Each shaper implements two burst modes. By default, a leaky bucket is continuously assigned new credit according to the configured shaper rate. This implies that during idle periods, credit is building up, which allows for a burst of data when there are again data to transmit. This is not convenient in an Audio/Video Bridging (AVB) environment where this behavior enforces a requirement for larger buffers in end-equipment. To circumvent this, each shaper can enable an AVB mode (SE_CFG.SE_AVB_ENA) in which credit is only assigned during periods where the scheduler element has data to transmit and is waiting for another scheduler element to finish a transmission. This AVB mode prevents the accumulation of large amount of credits.

The shapers can also operate as a dual leaky bucket designed to comply with MEF standards. By default, the dual leaky bucket shaper forwards frames with the committed information rate and holds back frames in the queue system when CIR is exceeded. However, the dual leaky bucket functionality allows the shaper to push traffic forward with a higher bandwidth (the excess information rate) if the queue system is filling up.

In addition to configuration of the committed information rate in CIR_CFG, the excess information rate can be configured as follows:

- Excess maximum excess rate – Specified in EIR_CFG.EIR_RATE in steps of 100 kbps. Maximum is 3.28 Gbps.
- Excess maximum excess burst size – Specified in EIR_CFG.EIR_BURST in steps of 4 kilobytes. Maximum is 252 kilobytes.

The excess information rate is used when certain fill levels in the queue system are exceeded.

Frames exceeding the committed maximum rate can optionally be colored yellow by setting EIR_CFG.EIR_MARK_ENA.

4.14.3 DEFICIT WEIGHTED ROUND ROBIN

The DWRR uses a cost-based algorithm compared to a weight-based algorithm. A high cost implies a small share of the bandwidth. DWRR is enabled when SE_CFG.SE_DWRR_CFG>0 and SE_CFG.RR_ENA = 0. The participating inputs are then inputs 0 through SE_CFG.SE_DWRR_CFG-1. Anything from 0 to 12 weighted inputs can participate.

Each input is programmed with a cost (SE_DWRR_CFG.DWRR_COST). A cost is a number between 1 and 32. The programmable DWRR costs determine the behavior of the DWRR algorithm. The costs result in weights for each input to the scheduler element. The weights are relative to one another, and the resulting share of the egress bandwidth for a particular input is equal to the input's weight divided by the sum of all the inputs' weights. The algorithm is byte-based and takes the frame lengths into account.

Costs are easily converted to weights and vice versa given the following two algorithms. The following algorithms are shown with six participating inputs but can be applied to other configurations as well.

Weights to Costs Given a desired set of weights (W0, W1, W2, W3, W4, W5), the costs can be calculated using the following algorithm.

1. Set the cost of the queue with the smallest weight (Wsmallest) to cost 32.
2. For any other queue Qn with weight Wn, set the corresponding cost Cn to
   Cn = 32 × Wsmallest/Wn

Costs to Weights: Given a set of costs for all queues ($C0$, $C1$, $C2$, $C3$, $C4$, $C5$), the resulting weights can be calculated using the following algorithm:

1. Set the weight of the queue with the highest cost (Chighest) to 1.
2. For any other queue Qn with cost Cn, set the corresponding weight Wn to Wn = Chighest/Cn

Cost and Weight Conversion Examples

Implement the following bandwidth distributions.

- Input 0: 5% ($W0$ = 5)
- Input 1: 10% ($W1$ = 10)
- Input 2: 15% ($W2$ = 15)
- Input 3: 20% ($W3$ = 20)
- Input 4: 20% ($W4$ = 20)

- Input 5: 30% (*W5* = 30)

Given the algorithm to get from weights to costs, the following costs are calculated:

- *C0* = 32 (Smallest weight)
- *C1* = 32∗5/10 = 16
- *C2* = 32*5/15 = 10.67 (rounded up to 11)
- *C3* = 32*5/20 = 8
- *C4* = 32*5/20 = 8
- *C5* = 32*5/30 = 5.33 (rounded down to 5)

Due to the rounding off, these costs result in the following bandwidth distribution, which is slightly off compared to the desired distribution:

- Input 0: 4.92%
- Input 1: 9.85%
- Input 2: 14.32%
- Input 3: 19.70%
- Input 4: 19.70%
- Input 5: 31.51%

4.14.4 ROUND ROBIN

The round robin (RR) uses a simple round robin algorithm where each participating input is served one after another. RR is enabled when SE_CFG.SE_DWRR_CFG>0 and SE_CFG.RR_ENA = 1. The participating inputs are then inputs 0 through SE_CFG.SE_DWRR_CFG-1. Anything from 0 to 12 weighted inputs can participate.

The RR algorithm is frame-based and does not take the frame lengths into account.

4.14.5 SHAPING AND DWRR SCHEDULING EXAMPLES

This section provides examples and additional information about the use of the egress shapers and scheduler. The following assumes a scheduler element is connected to the egress queues.

4.14.5.1 Mixing DWRR and Shaping Example

- Output from scheduler element is shaped down to 500 Mbps.
- Queues 7 and 6 are strict and queues 5 through 0 are weighted.
- Queue 7 is shaped to 100 Mbps.
- Queue 6 is shaped to 50 Mbps.
- The following traffic distribution is desired for queue 5 through 0:
  Q0: 5%, *Q1*: 10%, *Q2*: 15%, *Q3*: 20%, *Q4*: 20%, *Q5*: 30%.
- Each queue receives 125 Mbps of incoming traffic.

Table 4-34 lists the DWRR configuration and the resulting egress bandwidth for the various queues.

TABLE 4-34: EXAMPLE OF MIXING DWRR AND SHAPING

Queue	Distribution of Weighted Traffic	Configuration Costs/Weights (C*n*/W*n*)	Result: Egress Bandwidth
Q0	5%	32 / 1	1/(1+2+2.9+4+4+6.4) × (500 – Mbps – 150 Mbps) = 17.2 Mbps
Q1	10%	16 / 2	2/(1+2+2.9+4+4+6.4) × (500 – Mbps – 150 Mbps) = 34.5 Mbps
Q2	15%	11 / 2.9	2.9/(1+2+2.9+4+4+6.4) × (500 – Mbps – 50 Mbps) = 50.1 Mbps
Q3	20%	8 / 4	4/(1+2+2.9+4+4+6.4) × (500 – Mbps – 150 Mbps) = 68.9 Mbps
Q4	20%	8 / 4	4/(1+2+2.9+4+4+6.4) × (500 – Mbps – 150 Mbps) = 68.9 Mbps
Q5	30%	5 / 6.4	6.4/(1+2+2.9+4+4+6.4) × (500 – Mbps – 150 Mbps) = 110.3 Mbps
Q6			50 = Mbps
Q7			100 = Mbps
Sum:	100%		**500** = **Mbps**

4.14.5.2 Strict and Work-Conserving Shaping Example

- Output from scheduler element is shaped down to 500 Mbps.
- All queues are strict.
- All queues are shaped to 50 Mbps.
- Queues 6 and 7 are work-conserving (allowed to use excess bandwidth).
- All queues receive 125 Mbps of traffic each.

Table 4-35 lists the resulting egress bandwidth for the various queues.

TABLE 4-35: EXAMPLE OF STRICT AND WORK-CONSERVING SHAPING

Queue	Result: Egress Bandwidth
Q0	50 Mbps
Q1	50 Mbps
Q2	50 Mbps
Q3	50 Mbps
Q4	50 Mbps
Q5	50 Mbps
Q6	75 Mbps (Gets the last 25 Mbps of the 100 Mbps in excess not used by queue 7)
Q7	125 Mbps (Gets 75 Mbps of the 100 Mbps in excess limited only by the received rate)
Sum:	**500 Mbps**

4.14.6 TIME AWARE SHAPER

The device is designed to support Enhancements for Scheduled Traffic from IEEE 802.1Q-2018, in the following referred to as Time-Aware Shaper (TAS).

Network availability can be split up into different time slots, and different traffic classes can be allocated one or more of the time slots. By preventing lower-priority traffic from being scheduled in specific time slots, higher-priority traffic using these time slots is protected from congestion.

4.14.6.1 Basic Operation

TAS associates a gate state with each queue input to a scheduling element which can be either open or closed. Only queues associated with an open gate state can transmit frames. Gate states are changed by scheduling gate operations at specified periodic intervals. One gate operation consists in assigning either an open or closed state to each of the 8 inputs to one scheduling element.

For example, to reserve one in every 10 milliseconds to priority 7 traffic and leave the remaining 9 milliseconds to be shared among priority 0-6, the gate operations listed in Table 4-36 could be scheduled.

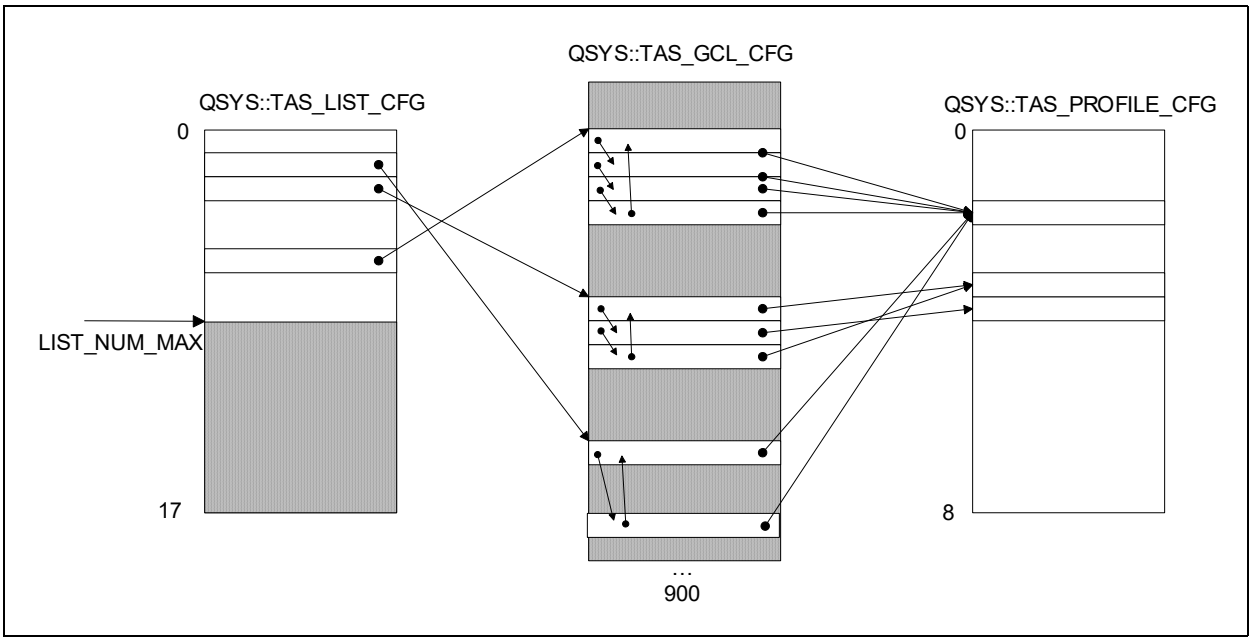
TABLE 4-36: TAS SCHEDULE EXAMPLE

Time (secs)	Close/Open Operation
0.000	Close priorities 0-6, open priority 7
0.001	Close priority 7, open priorities 0-6
0.010	Close priorities 0-6, open priority 7
0.011	Close priority 7, open priorities 0-6
0.020	Close priorities 0-6, open priority 7
0.021	Close priority 7, open priorities 0-6

In addition, a gate state operation can optionally trigger preemption of any preemptible frames currently being transmitted through the MAC, thereby quickly freeing up the port for transmission of express frames.

A TAS configuration consists of three main elements as shown in Figure 4-20.

FIGURE 4-20: TAS TABLES



The tables listed are described in more detail in the following sections.

- TAS_LIST_CFG. Typically one TAS list entry per port is used. It collects the list of gate operations to be scheduled periodically.
- TAS_GCL_CFG. Each TAS GCL entry defines one gate operation.
- TAS_PROFILE_CFG. Each TAS profile entry defines configuration shared between several gate operations, such as link speed.

To configure TAS, the following main steps should be taken:

1. Set QSYS::TAS_STATEMACHINE_CFG.REVISIT_DLY to 256 divided by the clock period in nanoseconds. This is required for proper operation. For example, value 40 is to be used for a 156,25 MHz clock frequency.
2. Set up TAS profiles as needed, for example one per port.
3. Allocate entries in TAS_GCL_CFG for each TAS list needed and write each gate operation entry into each TAS list.
4. Configure each TAS list entry needed and point it to the start of its list of gate operations in TAS_GCL_CFG.
5. Configure QSYS::TAS_CFG_CTRL.LIST_NUM_MAX to the highest TAS list entry in use.
6. Start each TAS list processing by writing its state to ADVANCING.

4.14.6.2 Gate Operations

A gate operation is specified using the following register fields:

- QSYS::TAS_GCL_CTRL_CFG.HSCH_POS
  Specifies the port for which the gate operation applies to
- QSYS::TAS_GCL_CTRL_CFG.GATE_STATE
  8 bits of open (1) or close (0) values to be set for the gate state
- QSYS::TAS_GCL_CTRL_CFG.OP_TYPE
  Specifies whether the gate operation is a Set, Set-And-Hold-MAC, or Set-And-Release-MAC.
- QSYS::TAS_GCL_CTRL_CFG2.PORT_PROFILE
  Reference to the profile to use for the gate operation
- QSYS::TAS_GCL_CTRL_CFG2.NEXT_GCL
  The address of the next gate operation in the list. The end of the list is identified by a gate operation with NEXT_GCL pointing to the first operation in the list.

In addition, QSYS::TAS_GCL_TIME_CFG.TIME_INTERVAL specifies the interval (in nanoseconds) until the next gate operation should be applied.

A gate operation specifies the state (open or closed) of each queue on a port. If preemption is enabled on the port, the operation can additionally set a MAC HOLD state on the port (or release a prior HOLD). When a MAC is in the HOLD state, any currently transmitting preemptible frame is immediately preempted if possible, and further preemptible frames are blocked from starting transmission until the HOLD is released.

Gate operations are allocated from a buffer with room for 900 operations. To configure a gate operation, first write QSYS::TAS_CFG_CTRL.GCL_ENTRY_NUM with the index of the operation, and then write the desired configuration into registers QSYS::TAS_GCL_CTRL_CFG, QSYS::TAS_GCL_CTRL_CFG2, and QSYS::TAS_GCL_TIME_CFG.

Gate operations form a linked list through the NEXT_GCL field. The first GCL is identified by a pointer in the TAS list entry, and the last GCL is identified by having a NEXT_GCL field that points back to the first entry. Allocation of gate operation is managed by the user application.

4.14.6.3 TAS List Scheduling

Up to 18 independent lists of TAS gate operations are available. A TAS list is associated with a linked list of gate operations. When the TAS list is operating, each gate operation is applied to the corresponding scheduling element at the start of its interval.

A TAS list is also associated with total cycle time. After the cycle time elapses, the TAS list starts over with the first gate operation. The cycle time must be at least as long as the sum of all gate operation intervals, and must also be less than one second.

Only one TAS list is processed in each clock cycle. This means there can be some delay from the nominal start of an interval until a gate operation is actually processed. The maximum delay in cycles is given by the number of lists being actively processed. This number is set in QSYS::TAS_CFG_CTRL.LIST_NUM_MAX. Note that a TAS list with higher index than this is not processed at all, regardless of other configurations. The minimal value for this is 24. Gate operation intervals must be configured long enough so that at most one interval can elapse between each time a TAS list is processed, that is it must be longer than LIST_NUM_MAX+1 cycles. Section 4.14.6.7, "Bandwidth and Latency Considerations" describes additional limits on gate operation timing.

To configure a TAS list, first write the index of the TAS list into QSYS::TAS_CFG_CTRL.LIST_NUM, and then write the appropriate values into the registers under the QSYS::TAS_LIST_CFG register group:

- TAS_BASE_TIME_NSEC, TAS_BASE_TIME_SEC_LSB, TAS_BASE_TIME_MSB
  This is the time at which to start processing the first gate operation in the TAS list.
- TAS_CYCLE_TIME_CONFIG
  The total cycle time for the TAS list.
- TAS_LIST_CFG.LIST_BASE_ADDR
  This is the index of the first gate operation in the TAS list thus specifying the start of the list.
- TAS_LIST_CFG.LIST_TOD_DOM
  This specifies the PTP domain (0, 1, or 2) used as time reference for TAS_BASE_TIME_*.
- TAS_STARTUP_CFG.OBSOLETE_IDX
  Optionally, this can be set to cause another TAS list to stop processing when this TAS list starts. This can be used to implement the distinction between the “Open” and “Admin” configurations of IEEE 802.1Q-2018. Setting TAS_STARTUP_CFG.OBSOLETE_IDX to the index of the TAS list itself causes the TAS list to start without affecting any other TAS lists.

The registers in the QSYS::TAS_LIST_CFG register group should not be modified unless QSYS::TAS_LIST_STATE.LIST_STATE is ADMIN (0), indicating the idle state of a TAS list.

To start processing of a TAS list, write QSYS::TAS_LIST_STATE.LIST_STATE to ADVANCING (1). The TAS list then moves to the PENDING state (2), and then into the OPERATING state (3) when the BASE_TIME is reached, at which point the gate operations in the TAS list become active.

A TAS list needs to spend at least one full cycle time in the PENDING state before it can move into the OPERATING state. For example, if the current time is 2.03 seconds and the cycle time is 0.05 seconds, the BASE_TIME should be set later than 2.08 seconds.

If the specified BASE_TIME is not sufficiently long ahead in the future, initial cycle time periods are skipped until the operation can begin at the start of a full cycle time. Note that a TAS list can only skip one cycle time each time it is processed. Thus, a TAS list should not be started at for example BASE_TIME=0, as that would require an excessive number of processing steps to skip forward to the current time. To avoid the overhead of skipping one cycle at a time, the

application should add an appropriate multiple of the cycle time before setting BASE_TIME. For example, if the current time is 2.03 seconds and the cycle time is 0.05 seconds, BASE_TIME should be set to 2.00 + 2*0.05 = 2.10 seconds rather than 2.00 seconds.

To stop a TAS list in the OPERATING state, a state of TERMINATING (4) can be written to QSYS::TAS_LIST_STATE.LIST_STATE. The TAS list then becomes idle in the ADMIN state when the end of a cycle time is reached. Since the TAS list is actively processing before being stopped, there is a race condition where the hardware may overwrite the TERMINATING state being written by the application. To handle this, it is necessary to read back the value of QSYS::TAS_LIST_STATE.LIST_STATE, and repeatedly rewrite the value of TERMINATING until a value of TERMINATING or ADMIN is read back.

A TAS list can also be stopped automatically from another TAS list that is replacing it, refer to Section 4.14.6.5, "Atomically Reconfiguring TAS Lists" for more details.

4.14.6.4 TAS Profiles

A TAS profile is associated with each gate operation through QSYS::TAS_GCL_CTRL_CFG.PORT_PROFILE. A TAS profile is typically set up for a specific egress port and shared by a list of gate operations. A maximum of 9 different TAS profiles can be configured.

A TAS profile determines two behaviors for a gate operation:

- Guard-banding, which initiates gate close operations at some configurable time in advance to ensure that all in-progress frames have time to complete their transmission before the nominal close time.
- holdAdvance, which is a configurable time in advance of the nominal Set-And-Hold-MAC time that a HOLD is signaled to a MAC, to ensure that there is sufficient time to preempt any actively transmitting preemptible frame.

IEEE 802.1Q-2018 requires that at the nominal time of a gate operation, all frames must have completed transmission if they belong to a traffic class that is closed by the operation. To ensure this, it is necessary to stop the scheduling of frames some time in advance so that any already-scheduled frames have time to complete. This time is called the guard band.

The required guard band time depends on several factors, including:

- The maximum size of a frame of the given traffic class that can currently be in the process of being transmitted.
- The normal latency for a frame through the various blocks in the port module until it is transmitted from the MAC.

The required guard band time can be reduced if preemption is enabled by using a Set-And-Hold-MAC gate operation. This causes any currently transmitting preemptible frame to be preempted, reducing the latency before the port is ready to transmit express frames:

- The latency of any currently transmitted frame is reduced to two minimum-sized frames as this is the limit of preemption.
- The latency from disassembler watermarks is eliminated since preemptible frames are blocked in the buffers while HOLD MAC is in effect.

To configure the guard-band time for traffic class P in TAS profile N, the following fields should be written:

- QSYS:TAS_PROFILE_CFG[N]:TAS_PROFILE_CONFIG.LINK_SPEED
  This sets the speed of the link: 10 Mbps, 100 MBps, 1 Gbps, or 2.5 Gbps.
- QSYS:TAS_PROFILE_CFG[N]:TAS_QMAXSDU_CFG[P].QMAXSDU_VAL
  This sets the guard band time in units of minimum-sized frames.
- QSYS:TAS_PROFILE_CFG[N]:HOLDADVANCE
  This sets the holdAdvance time in units of minimum-sized frames.

The actual guard band time becomes (64*8) bits * QMAXSDU_VAL / LINK_SPEED.

To control which traffic classes the guard banding applies to, the field QSYS::TAS_PROFILE_CONFIG.SCH_TRAFFIC_QUEUES is used. A class with scheduled traffic is one for which the corresponding bit is set to 1. Then:

- If a gate operation closes a not-scheduled traffic class, the close operation takes place a guard band time earlier than the nominal time for the operation.
- If the QSYS::TAS_CFG_CTRL.ALWAYS_GUARD_BAND_SCH_Q field is set, then all gate close operations happen a guard band time earlier, regardless of scheduled or non-scheduled classification.

There is no guard band time applied to a MAC RELEASE.

4.14.6.5 Atomically Reconfiguring TAS Lists

It is possible to switch TAS lists from one configuration to another in a seamless way, without an idle period between stopping the old configuration and starting the new configuration. This can be used to implement the distinction between “Oper” and “Admin” configuration parameters.

To atomically replace list A with list B, set HSCH::TAS_STARTUP_CFG::OBSOLETE_IDX of list B to the index of list A. As B is starting it will then signal for A to stop. By configuring correctly the timing of B relative to A, it is possible to ensure that B will take over at the precise time that A stops. This timing is controlled by HSCH::TAS_STARTUP_CFG::START-UP_TIME.

The list A will always complete a cycle before stopping. The stop happens just after the time of the last gate operation in a cycle, at which point A will automatically move into the idle ADMIN state. Thus, to stop A at the end of a particular cycle, the signal from B must arrive between the last gate operation of the previous cycle and the last gate operation of the current cycle.

The list B sends the stop signal to A at the time 256*STARTUP_TIME nanoseconds before B begins its first cycle. TAS lists are processed round-robin, one TAS list each clock cycle, so there can be a few cycles of jitter between the configured time and actual time.

A simple way to ensure correct timing is to set STARTUP_TIME of B to the difference between the start of the first cycle of B and the start of the last cycle of A:

STARTUP_TIME := first_cycle_start(B) - last_cycle_start(A)

This leaves ample time for the signal to arrive at the correct point, assuming that gate operation durations are longer than the maximum jitter in list processing.

The start of list B should preferably coincide with the stop of list A to ensure a seamless switch.

If the first cycle of list B starts later than the end of the last cycle of list A, then there will be an interval where no list is operating on the port. In this interval, the gate state of the port will be as set by the last gate operation in A.

If the first cycle of list B starts before the end of the last cycle of list A, then there will be an interval where both lists are operating on the same port. This will cause the gate operations of one list to overwrite the state set by a gate operation of the other list in the interval of overlap.

4.14.6.6 Port State after Stopping a TAS List

When a gate operation closes a queue for a port, that queue is permanently blocked until it opens again. Similarly, a MAC HOLD on a port remains in effect until another gate operation does a MAC RELEASE.

When a TAS list stops, the state of ports is left from the last gate operation in the list. The queue state can afterwards be set manually by writing to QSYS:TAS_LIST_STATE.LIST_STATE (after setting the port in QSYS:TAS_GATE_STATE_CTRL.HSCH_POS). This can be used to ensure that all queues are open after stopping the TAS list. It can also be used to set an initial state prior to the start time of a new TAS list.

A MAC HOLD left after stopping a list can be cleared by writing register SYS:PSTATE[N]:FPORT_STATE.MAC_HOLD.

4.14.6.7 Bandwidth and Latency Considerations

One TAS list is processed each clock cycle. This means that there is some variable latency from the nominal time of a gate operation until the operation actually takes effect. The number of cycles can be up to (QSYS::TAS_CFG_C-TRL.LIST_NUM_MAX + 1) or 18, whichever is higher. This latency is usually small compared to other latencies in the system related to frame transmission.

The duration of a gate operation must be larger than the time to process all active TAS lists since at most one gate operation is processed each cycle. At 156,25 MHz and with all 18 TAS lists active, this is just 18*6.4 = 115 nanoseconds, so this is normally not a real limitation.

4.15 Automatic Frame Injector

The automatic frame injector (AFI) provides mechanisms for periodic injection of PDUs.

The following blocks are involved in frame injection.

- QSYS: The frames are sent into the queue system by the CPU and stored in the queue system until deleted by software.
- AFI: Using timers and other parameters, AFI controls the automatic frame injection.
- REW: For outbound data path injections, the REW performs frame modifications.

AFI supports two types of automatic frame injection: timer triggered (TTI) and delay triggered (DTI).

- Timer Triggered Injection (TTI): Frame injection rate is controlled by timers with typical values of approximately 3 ms or higher. Only single-frame injections are supported; injection of sequences of frames is not supported.

For each frame, jitter can be enabled for the associated injection timer.

Timer triggered injection can be combined with delay triggered injection.

- Delay Triggered Injection (DTI): A sequence of frames, including configurable delay between frames, is injected.

AFI supports up to 4 active DTI frame sequences at a time.

4.15.1 INJECTION TABLES

The main tables used to configure frame injection are frame table, DTI table, and TTI table.

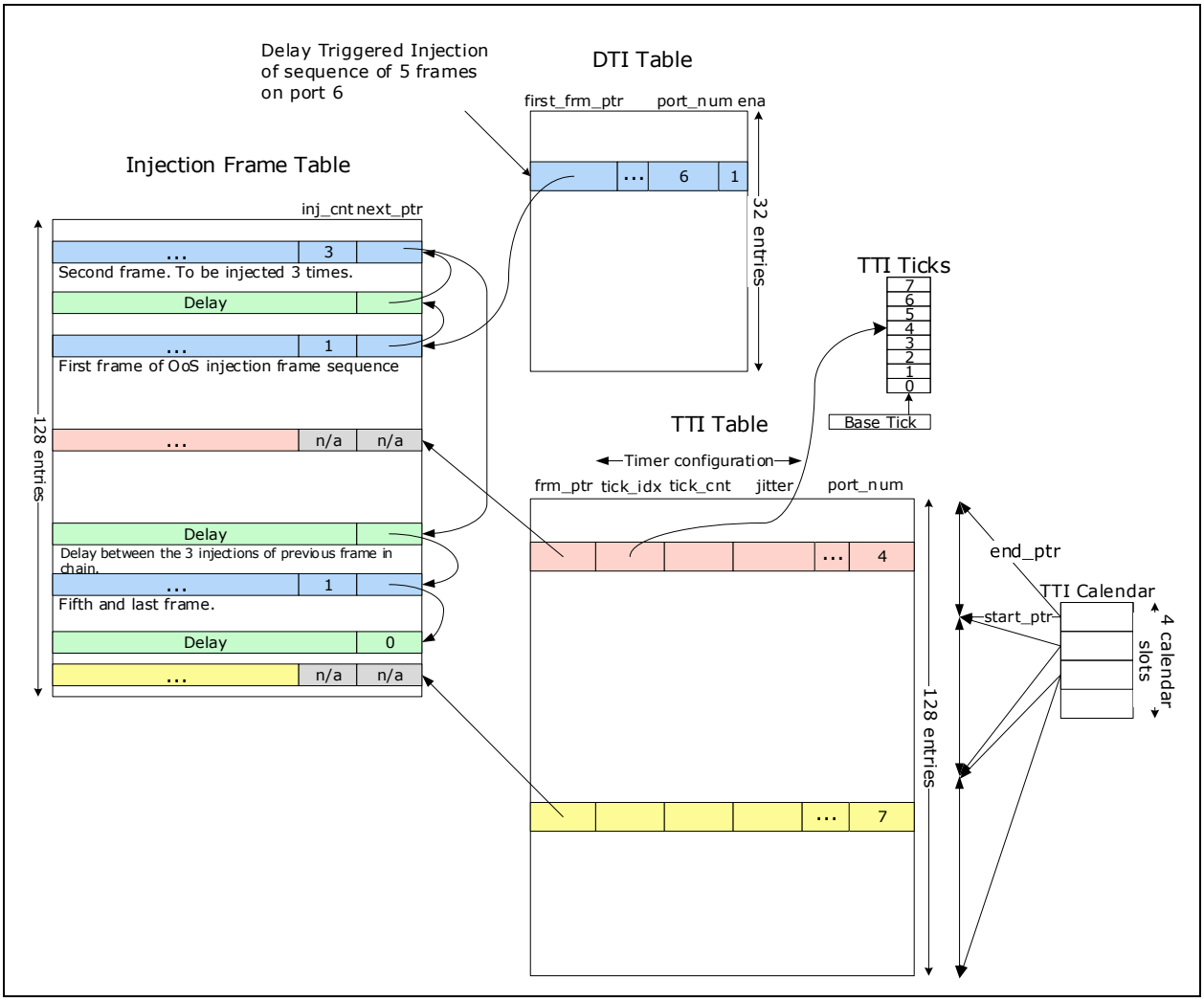
Injection frames for both TTI and DTI are configured in the frame table. The frame table itself does not hold the frame, but rather a pointer to where the frame is located in the buffer memory within QSYS.

For DTI, frames can be configured into a linked list of frames to be injected. Each frame in the linked list may be configured to be injected multiple times before proceeding with the next frame in the chain. Each frame in the sequence is followed by one or more delay entries, specifying the delay between injections.

Timers for each TTI frame are configured in the TTI table. TTIs are always single frame injections, so the inj_cnt and next_ptr fields in the frame table are unused for TTI. Delay entries are not used for TTI.

The TTI calendar controls the frequency with which different parts of the TTI table are serviced.

FIGURE 4-21: INJECTION TABLES



4.15.2 FRAME TABLE

4.15.3 DELAY TRIGGERED INJECTION

Delay triggered injection supports up to 4 flows at a time. For each DTI flow, the DTI table holds a pointer to a sequence of frames and delay entries, as well as the queue and port number, into which the frames are injected. The NEXT_PTR field of the frame table is used to link frames and delay entries together into a sequence. The frame sequence can have any length, only limited by the size of the frame table.

Each DTI sequence can be configured to be injected repeatedly until stopped, or injected a specified number of times. For example, if the frame sequence consists of five frames with INJ_CNT = 1, and the sequence is configured to be injected 1000 times, then a total of 5000 frames will be injected.

DTI can be used in conjunction with TTI, for example, using TTI to inject OAM PDUs for measuring performance metrics of the service.

Frame entries in a DTI sequence may point to the same frame in the queue system's buffer memory. For example, as shown in Figure 4-22, frame A and frame B shown may point to the same frame in the buffer memory.

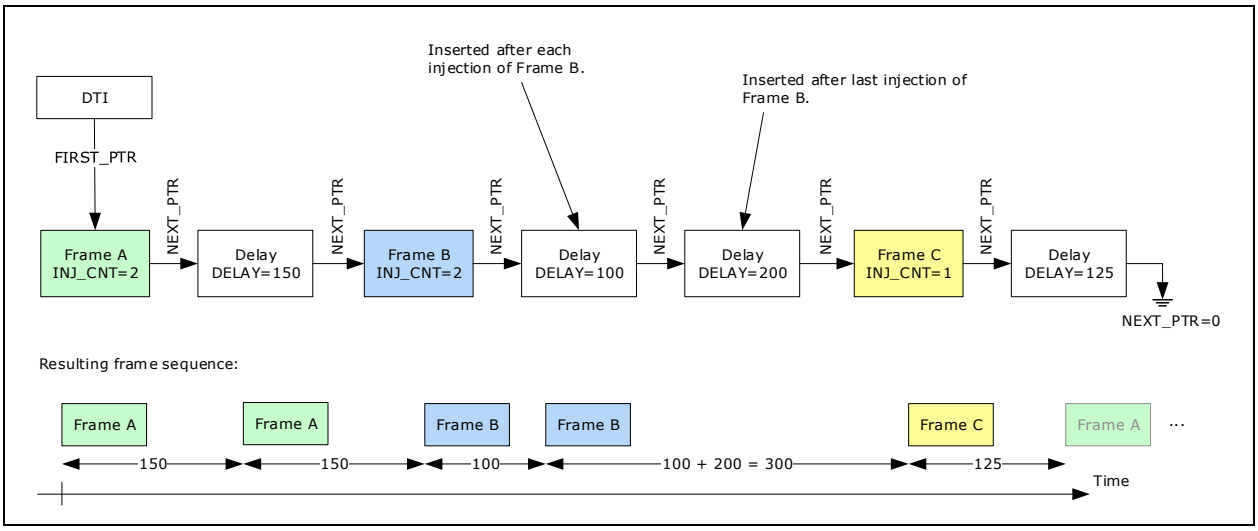
The delay entries in a DTI sequence must be configured such that they are longer than the transmission time (given the port speed) of the preceding frame in the sequence.

If a DTI sequence is configured with two consecutive frame entries, then these will be injected with a minimum delay of approximately 14 clock cycles.

4.15.3.1 Frame Delay Sequence

A DTI flow typically consists of a sequence of alternating frame/delay entries in the frame table. If a given frame is to be injected multiple times (FRM_ENTRY_PART0.INJ_CNT>1), then the delay of the following entry in the sequence is applied after each injection. If additional delay is required after the last injection, then this can be achieved by an additional delay entry. The use of delay entries is shown in Figure 4-22.

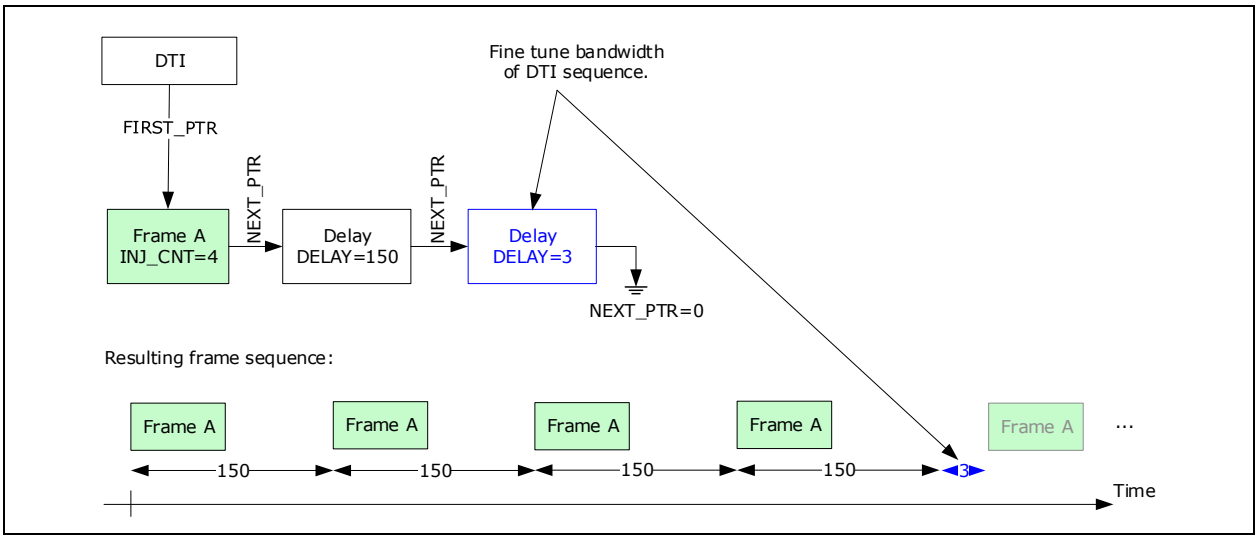
FIGURE 4-22: DTI FRAME/DELAY SEQUENCE EXAMPLE



4.15.3.2 Bandwidth Fine Tuning

The combination of INJ_CNT and two following delay entries can be used to fine tune the rate generated by a DTI flow. This is shown in Figure 4-23, where a small delay after the four injections of frame A is used to fine tune the rate of the DTI flow.

FIGURE 4-23: FINE TUNING BANDWIDTH OF DTI SEQUENCE



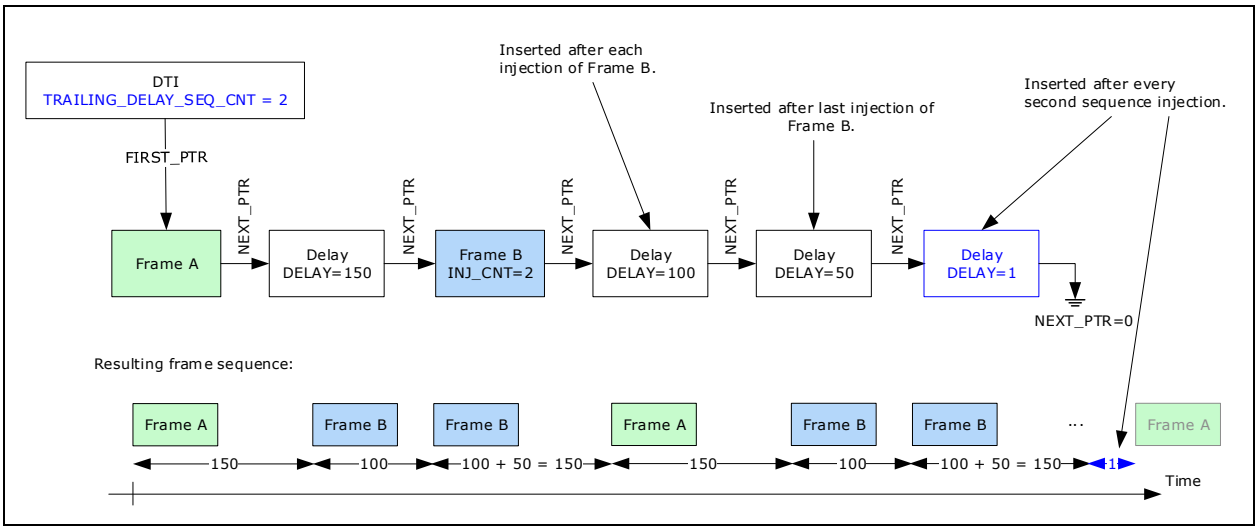
By combining the setting of INJ_CNT and the DELAY value of the last delay depicted, the bandwidth of the DTI sequence can be controlled with high granularity, because the additional (blue) delay will only be applied for every INJ_CNT injections of frame A.

The bandwidth adjustment mechanism shown may be insufficient for controlling the bandwidth of multi-frame sequences (sequences with different frames), because the last delay will be applied for every injection of the frame sequence.

4.15.3.3 Fine Tuning Bandwidth of Multiframe Sequence

For multiframe sequences, additional fine tuning of the bandwidth of the entire sequence can be achieved by configuring the last delay in the sequence, also known as the “trailing delay”, to only be applied for every Nth injection of the sequence. This is achieved using the TRAILING_DELAY_SEQ_CNT parameter and is shown in Figure 4-24.

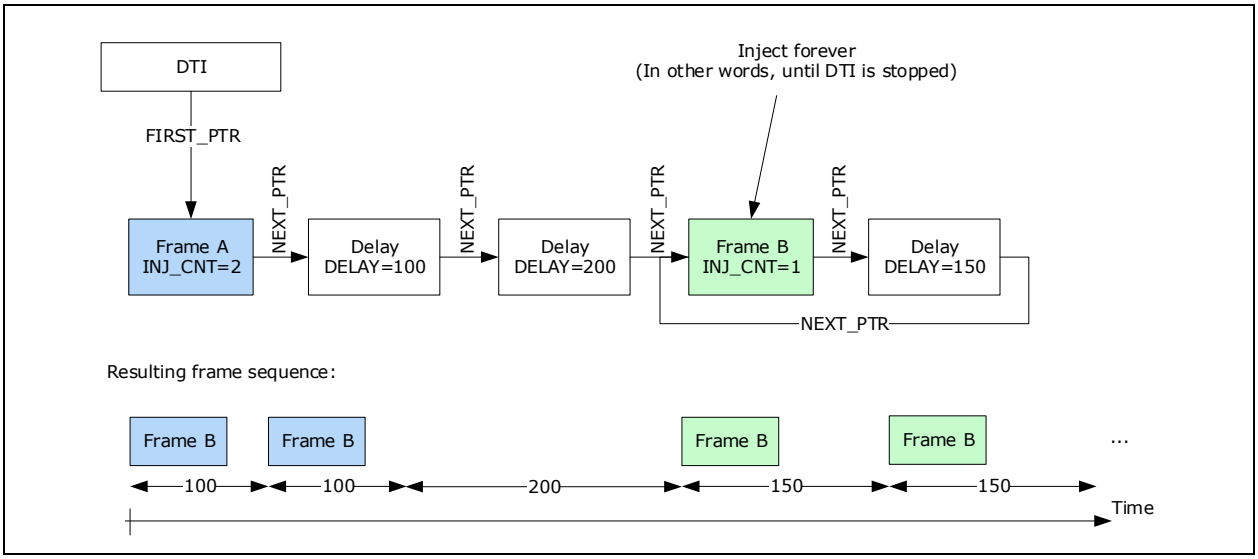
FIGURE 4-24: FINE TUNING BANDWIDTH OF MULTIFRAME DTI SEQUENCE



4.15.3.4 Burst Size Test Using Single DTI

If a NEXT_PTR is set to point back to a previous entry in the sequence, then injection will continue forever, that is, until the DTI is stopped. This is depicted in Figure 4-25. The configuration shown can be used for leaky bucket burst size testing, where the first part of the sequence empties the bucket, followed by a steady frame flow with the rate of the leaky bucket.

FIGURE 4-25: DTI FRAME/DELAY SEQUENCE USING INJECT FOREVER

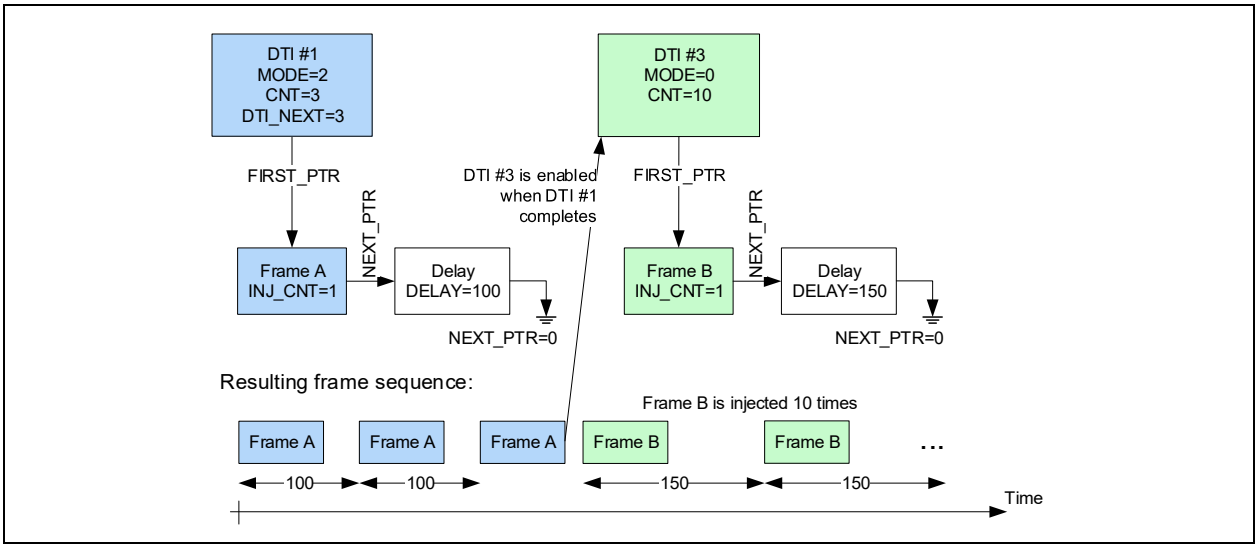


4.15.3.5 Burst Size Test Using DTI Concatenation

Two (or more) DTIs can be concatenated, such that when one DTI completes, another DTI is activated. This is depicted in the following illustration. It can be used for leaky bucket burst size testing, where the first part of the sequence empties the bucket, followed by a steady frame flow with the rate of the leaky bucket.

Note that when DTI concatenation is used, the last delay entry in the frame-delay sequence of the first DTI is not awaited before starting the next DTI. That is, the next DTI is started when the first DTI reads an entry with NEXT_PTR = 0. As shown in Figure 4-26, the delay between the last Frame A and the first Frame B will not be 150 clock cycles, but instead approximately 12 clock cycles. If this is considered problematic, it can be addressed by using an additional DTI with a sequence consisting only of two delay entries. For example, DTI #2 could be inserted between DTI#1, and DTI#3 and DTI#2 could then be configured with a sequence consisting of two delay entries, with the first having delay = 150.

FIGURE 4-26: DTI CONCATENATION



4.15.3.6 DTI Bandwidth

The maximum bandwidth supported for DTI flows depend on

- Clock frequency
- Frame size
- Available bandwidth on the cell bus (that is, port configuration dependent)

For calculations based on clock frequency and frame size, see the description of AFI:DTI_TBL in the register list. The bandwidth, which a DTI is injecting into a queue, must not exceed the bandwidth, which HSCH is transmitting from the queue, since then the port's FRM_OUT_MAX will then be reached, thus possibly affecting other DTIs injecting on the same port. The only exception to this is, if there is only one DTI injecting on the port and thus no other DTIs, which can be affected.

4.15.3.7 DTI Sequence Processing Time

For high bandwidth DTI flows, the time it takes to process the delays and frames in a DTI sequence must be taken into account:

- Processing a frame entry with no succeeding delay entry takes 12 clock cycles.
- Processing a frame entry with a succeeding delay entry takes 12 clock cycles.
- Processing a delay entry, which does not succeed a frame entry takes 12 clock cycles.

For example, a sequence consisting of a frame entry followed by two delay entries will take 24 clock cycles to process. If the frame is a 64-byte frame, and if the clock period is 4 ns, this will correspond to a maximum flow bandwidth of approximately 7 Gbps ((64 + 20) × 8 bits/(24 × 4 ns) = 7 Gbps), regardless of the delay values configured for the two delays.

Although it takes 12 clock cycles to process a delay entry, it is valid to configure a delay entry with a delay smaller than 12 clock cycles, because this will be compensated for when generating the delay between later frames in the sequence. For more information, see Figure 4-24. For example, the delay between the rightmost frame B and the succeeding frame A will be 150 + 12 clock cycles (instead of 150 + 1). To compensate for this, however, the delay between the succeeding two frames A will be 150 – 11 clock cycles.

4.15.4 TIMER TRIGGERED INJECTION

For TTIs, frame injection is controlled by timers. To control the time between injections, configure the following registers for each TTI:

- TTI_TIMER.TICK_IDX. For each TTI, one of eight configurable timer ticks shown in the following table must be chosen.
- TTI_TIMER.TIMER_LEN. The number of timer ticks that must elapse between each injection.

That is, the period between each injection becomes tick_period × TIMER_LEN.

An entry in the TTI table is checked approximately every two clock cycles. If the TTI's tick (configured through TICK_IDX) has changed, then the TICK_CNT is decremented. If it becomes zero, the frame is injected, and TICK_CNT is set to TIMER_LEN.

4.15.4.1 TTI Calendar

In default configuration, TTI table entries are serviced from 0-127, then restarted again at entry 0.

When configuring the TTI table, the following must be considered.

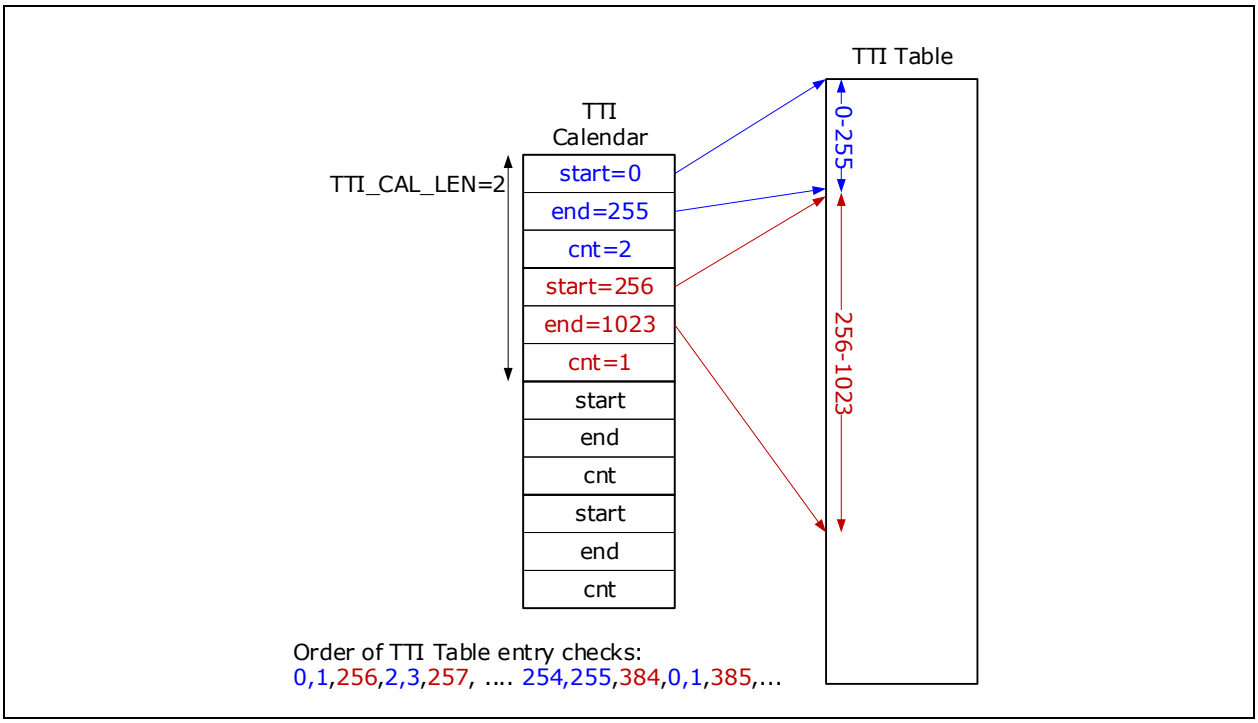
- It takes up to 4 clock cycles, worst-case, to service an entry in the TTI table.
- A TTI must be serviced more frequently than the frequency of the TTI's tick (see TTI_TIMER.TICK_IDX).

This means that looping through the entire TTI table may take 128 × 4 = 512 clock cycles. With a clock period of for instance 4 ns, this corresponds to approximately 2 µs. In other words no TTI must use a tick faster than 2 µs.

If TTIs faster than 2 µs are required, the TTI calendar can be used to have some entries in the TTI table serviced more often than others, meaning the TTI table can be split into sections with TTIs using ticks of different speeds.

Figure 4-27 shows an example of a TTI calendar. In this example, the TTI table has 1024 entries.

FIGURE 4-27: TTI CALENDAR EXAMPLE



In the example, it takes (256/2) × (1 + 2) × 4 = 1536 clock cycles to service all TTIs in the blue section. With a clock cycle for instance 4 ns, the TTIs in the blue section must not use ticks faster than 1536 × 4 = 6144 ns.

Similarly, the 768 TTIs in the red section must not use ticks faster than
(768/1) × (1 + 2) × 4 × 4 ns = ~36.9 µs

4.15.4.2 TTI Table Update Engine (AFI TUPE)

The AFI Table Update Engine (AFI TUPE) can be used to quickly update a large number of entries in the TTI Table such as disabling or enabling timers in fail-over scenarios.

The following TTI parameters can be used as part of the TUPE criteria.

- PORT_NUM
- QU_NUM
- TUPE_CTRL

The parameters prefixed “CMD” specify the commands, which are applied to any TTIs, which fulfill the TUPE criteria. Using the TUPE command parameters, the following TTI parameters can be changed.

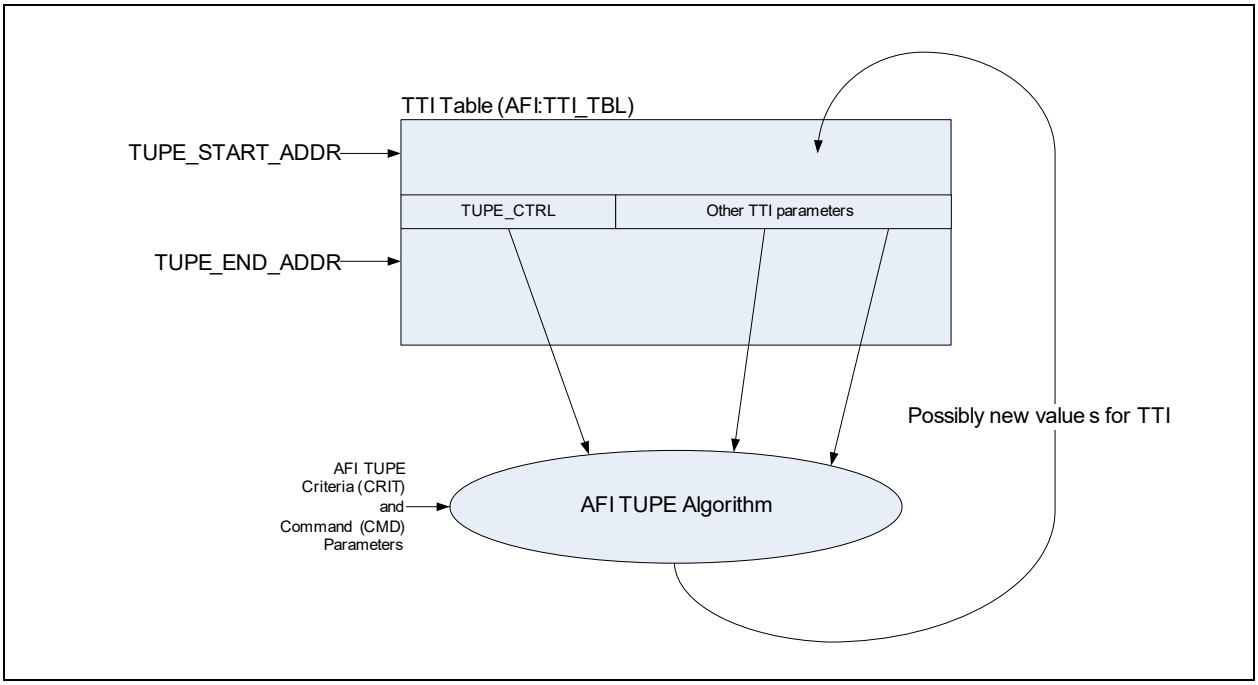
- TIMER_ENA
  Timers can be enabled/disabled.
- PORT_NUM
  Injections can be moved to a different port.
- QU_NUM
  Injections can be moved to a different queue.

While TUPE is running no injections will be performed for any TTIs which match the configured TUPE criteria and fall within the configured TUPE address range.

The TUPE_CTRL field in the TTI Table can be used to classify TTIs into different groups, such that a TUPE command can easily address all TTIs within such group.

Figure 4-28 shows the AFI TUPE functionality.

FIGURE 4-28: AFI TUPE



The full algorithm used by AFI TUPE is shown in the following pseudo code. Configuration parameters are prefixed “csr.”

```
// AFI Table UPdate Engine (AFI TUPE) Algorithm
```

```
bool tupe_ctrl_match;
for (addr = csr.tupe_start_addr; addr < csr.tupe_end_addr; addr++) {
    tti_tbl_entry = tti_tbl[addr];

    tupe_ctrl_match = FALSE;
    for (i = 0; i < 2; i++) {
        if ((tti_tbl_entry.tupe_ctrl & csr.crit_tupe_ctrl_mask) ==
            csr.crit_tupe_ctrl_val[i]) {
            tupe_ctrl_match = TRUE;
        }
    }
    if (
        (// Check matching TUPE_CTRL value
        tupe_ctrl_match)
        &&
        (// If enabled, check if TTI's PORT_NUM matches csr.crit_port_num_val
        !csr.crit_port_num_ena
        ||
        (tti_tbl_entry.port_num == csr.crit_port_num_val))
        &&
        (// If enabled, check if TTI's QU_NUM matches csr.crit_qu_num_val
        !csr.crit_qu_num_ena
        ||
        (tti_tbl_entry.qu_num == csr.crit_qu_num_val))
    ) {
    // TTI fulfills criterias => Apply TUPE command to TTI
    // timer_ena
    if (csr.cmd_timer_ena_ena) {
        tti_tbl_entry.timer_ena = csr.cmd_timer_ena_val;
    }
    // port_num
    if (csr.cmd_port_num_ena) {
        tti_tbl_entry.port_num = csr.cmd_port_num_val;
    }
    // qu_num
    if (csr.cmd_qu_num_ena) {
        tti_tbl_entry.qu_num = csr.cmd_qu_num_val;
    }
    // Write back to TTI table
    tti_tbl[addr] = tti_tbl_entry;
    }
}
```

4.15.5 INJECTION QUEUES

DTI and TTI may inject into any egress queue in the queue system. In injection context, the queues can be divided into the following categories:

- Normal queues: These queues are processed by each level of SEs in HSCH. Frames injected into such queues are thus subject to the M-DWRR and shaping algorithms of the SEs. Frames are injected into the tail of the selected queue.
- Port injection queue without shaping: These queues, one for each port, inject frames on the port with higher priority than normal queues. The state of the DLB shaper is disregarded and is not updated with the size of the transmitted frame.
  Injected frames are injected into the tail of the queue, but due to the high priority, the queue will normally be (almost) empty.
- Port injection queue with shaping: These queues, one for each port, inject frames on the port with higher priority than normal queues. The state of the port shaper is respected and is updated with the size of the transmitted frame.
  Injected frames are injected into the tail of the queue, but due to the high priority, the queue will normally be (almost) empty.

The queue into which a TTI or DTI injects frames is controlled by the QU_NUM parameter. The actual value to be used for this parameter depends on the HSCH configuration.

4.15.6 ADDING INJECTION FRAME

To add a frame for injection by AFI, the CPU must follow this procedure:

1. Set AFI::MISC_CTRL.AFI_ENA to 1 before using AFI.
2. Send the frame to AFI. AFI_INJ must be set in the IFH.
3. Poll AFI::NEW_FRM_CTRL.VLD to await the frame being received by AFI.
4. When the frame has been received by AFI, then copy frame information from AFI::NEW_FRM_INFO to an unused entry in the Frame table.
5. Clear AFI::NEW_FRM_CTRL.VLD.
6. TTI: Set up the TTI table to have the frame injected with the desired interval and to the desired queue and port. For more information, see Section 4.15.4, Timer Triggered Injection.
7. DTI: For injection of a sequence of frames, repeat steps 1 through 4

Set up the DTI Table to inject the frames.

Configure other DTI parameters.

4.15.7 STARTING INJECTION

TTI is enabled by using TTI_INIT to initialize and then setting TTI_ENA = 1.

Each TTI is started by setting TIMER_LEN a non-zero value.

Each DTI is started by setting DTI_CTRL.ENA = 1.

4.15.8 STOPPING INJECTION

Each TTI is stopped by setting TIMER_LEN to 0.

All TTIs can be stopped by setting TTI_ENA = 0.

Each DTI is stopped by setting DTI_CTRL.ENA = 0.

4.15.9 REMOVING INJECTION FRAMES

This section provides information about removing a single frame (TTI) or frame sequence (DTI) from the buffer memory.

4.15.9.1 Single Frame (TTI)

To remove a single frame from buffer memory, the frame must be injected one more time. This “removal injection” does not result in a frame transmission, but purely serves to remove the frame from the buffer memory.

Use the following procedure to remove a single frame from buffer memory:

1. Set the FRM_RM bit for frame in the frame table.
2. For TTI: Set TIMER_LEN to 0x1ff.
3. Poll the FRM_GONE bit until it gets set by AFI.

When performing removal injection, injection must be enabled for the corresponding port by setting AFI:PORT_TBL[<port>]:PORT_CFG.FRM_OUT_MAX != 0.

4.15.9.2 Frame Sequence (DTI)

Use the following procedure to remove a DTI frame sequence from buffer memory:

1. Disable the DTI by setting DTI_CTRL.ENA = 0.
2. Set the FRM_RM bit for each frame to be removed in the frame table.
3. Set DTI_FRM.NEXT_FRM_PTR to DTI_FRM.FIRST_FRM_PTR.
4. Set DTI_MODE.MODE = 0.
5. Set DTI_CNT.CNT = 1.
6. Set DTI_MODE.FRM_INJ_CNT to 0.
7. Optionally, set all delays in sequence to 0 to speed up the removal procedure.
8. Set DTI_CNT_DOWN.CNT_DOWN to 0.
9. Set DTI_CTRL.ENA to 1 to enable the DTI.
10. Poll the FRM_GONE for the last frame to be removed until the bit gets set by AFI.

This procedure causes the frames to be injected one last time and through this, be removed from buffer memory. The frames will not actually be transmitted on the destination port.

When performing removal injection, injection must be enabled for the corresponding port by setting AFI:PORT_TBL[<port>]:PORT_CFG.FRM_OUT_MAX != 0.

4.15.10 PORT PARAMETERS

To ensure that the queue system does not overflow with injected frames, for example, if a port is in flow control, an upper bound on the number of injected, but not yet transmitted, frames per port can be configured in FRM_OUT_MAX.

If FRM_OUT_MAX is reached, then any DTI injections are postponed until the number of outstanding frames falls below FRM_OUT_MAX. TTI injections uses a slightly higher limit of FRM_OUT_MAX+TTI_FRM_OUT_MAX. This ensures that TTI injection are still possible, even if a DTI flow faster than the port speed has been configured.

If PFC is used in conjunction with automatic frame injection, then either all frames must be injected into queues controlled by the same flow control priority, or all frames must be injected into queues that cannot be stopped by PFC.

If EEE is used in conjunction with automatic frame injection, then all frames for that port must be injected into queues with the same urgent configuration.

Setting FRM_OUT_MAX = 0 will stop all injections on port.

4.16 Rewriter

The device includes a rewriter common for all ports that determines how egress frames are edited before transmission. The rewriter performs the following editing:

- VLAN editing; tagging of frames and remapping of PCP and DEI.
- Redundancy tagging.
- DSCP remarking; rewriting the DSCP value in IPv4 and IPv6 frames based on classified DSCP value.
- FCS updating.
- Precision Time Protocol time stamp updating.
- EtherCAT time stamp updating.
- OAM, MRP, and DLR updating.

Each port including the CPU ports has its own set of configuration in the rewriter. Each frame is handled by the rewriter one time per destination port.

4.16.1 VLAN EDITING

The rewriter performs five steps related to VLAN editing for each frame and destination:

1. VLAN popping - Zero, one, or two VLAN tags are popped from the frame.
2. ES0 lookup - ES0 is looked up for each of the frame's destination ports. The action from ES0 controls the pushing of VLAN tags.
3. VLAN push decision - Deciding the number of new tags to push and which tag source to use for each tag. Tag sources are: Port and ES0 (tag A and tag B).
4. Constructing the VLAN tags - The new VLAN tags are constructed based on the tag sources' configuration.
5. VLAN pushing - the new VLAN tags are pushed.

4.16.1.1 VLAN Popping

The rewriter initially pops the number of VLAN tags specified by the VLAN_POP_CNT parameter received with the frame from the classifier or VCAP IS1. Up to two VLAN tags can be popped. The rewriter itself does not influence the number VLAN tags being popped.

4.16.1.2 ES0 Lookup

For each of the frame's destination ports, VCAP ES0 is looked up using the ES0 key. The action from an ES0 hit is used in the following to determine the frame's VLAN editing.

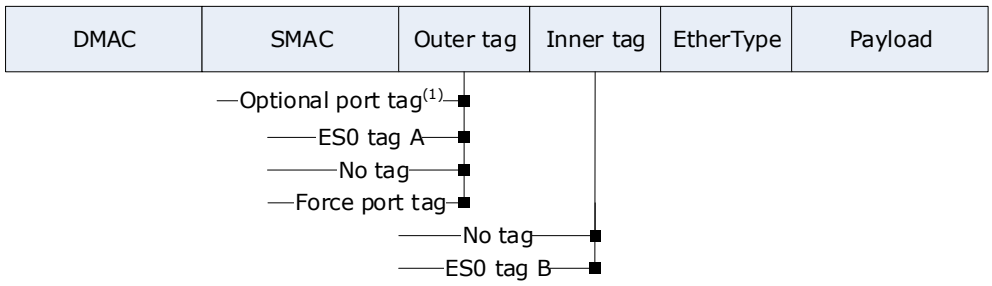
4.16.1.3 VLAN Push Decision

After popping the VLAN tags, the rewriter decides whether to push zero, one, or two new VLAN tags to the outgoing frame according to the port's tagging configuration in register TAG_CFG and the action from a potential VCAP ES0 hit. The up to two tags can originate from either the port (port tag) or ES0 (ES0 tag A and ES0 tag B).

By default, the port can push one tag according to the port's configuration in TAG_CFG. If the ES0 lookup results in an entry being hit, the ES0 action can overrule the port configuration and push two tags by itself (ES0 tag A and ES0 tag B) or it can combine port tagging and ES0 tagging.

Figure 4-29 shows an overview of the available tagging options.

FIGURE 4-29: TAGGING OVERVIEW



(1) Port tag is controlled by REW:PORT:TAG_CFG.TAG_CFG.

Table 4-37 lists all combinations of port tagging configuration and ES0 actions that control the number of tags to push and which tag source to use (port, ES0 tag A, or ES0 tag B):

TABLE 4-37: TAGGING COMBINATIONS

ES0_ACTION	TAG_CFG	Tagging action
No ES0 hit	Controls port tag	Port tag is pushed according to TAG_CFG as outer tag. Available options are: Tag all frames with port tag. Tag all frames with port tag, except if classified VID=0 or classified VID=PORT_VLAN.PORT_VID. Tag all frames with port tag, except if classified VID=0 No port tag No inner tag.
PUSH_OUTER_TAG=0 PUSH_INNER_TAG=0	Controls port tag	Port tag is pushed according to TAG_CFG as outer tag. Available options are: Tag all frames with port tag. Tag all frames with port tag, except if classified VID=0 or classified VID=PORT_VLAN.PORT_VID. Tag all frames with port tag, except if classified VID=0 No port tag No inner tag.
PUSH_OUTER_TAG=1 PUSH_INNER_TAG=0	Don't care	ES0 tag A is pushed as outer tag. No inner tag.
PUSH_OUTER_TAG=2 PUSH_INNER_TAG=0	Don't care	Port tag is pushed as outer tag. This overrules port settings in TAG_CFG. No inner tag.
PUSH_OUTER_TAG=3 PUSH_INNER_TAG=0	Don't care	No tags are pushed. This overrules port settings in TAG_CFG.

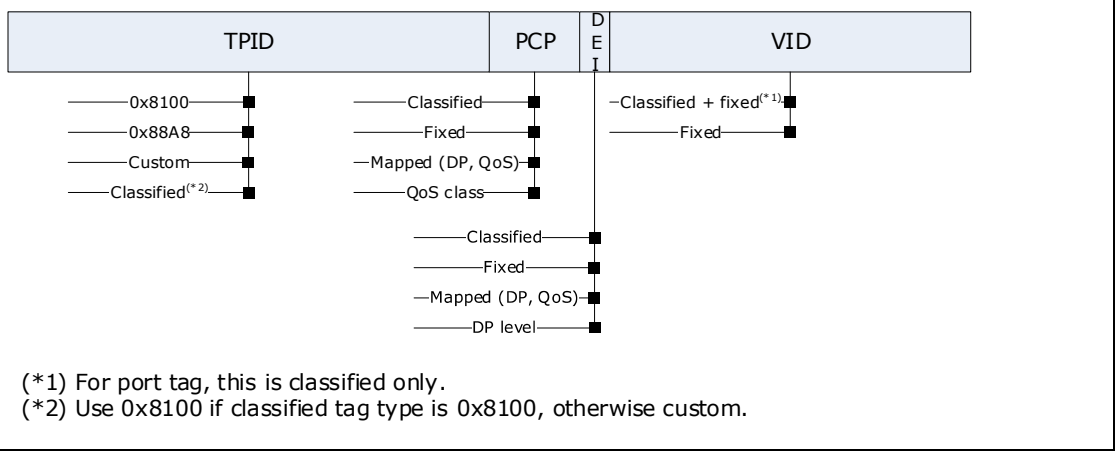
TABLE 4-37: TAGGING COMBINATIONS (CONTINUED)

ES0_ACTION	TAG_CFG	Tagging action
PUSH_OUTER_TAG=0 PUSH_INNER_TAG=1	Controls port tag	Port tag is pushed according to TAG_CFG as outer tag. The following options are available: Tag all frames with port tag. Tag all frames with port tag, except if classified VID=0 or classified VID=PORT_VLAN.PORT_VID. Tag all frames with port tag, except if classified VID=0 No port tag ES0 tag B is pushed as inner tag. ES0 tag B is effectively the outer tag if the port tag is not pushed.
PUSH_OUTER_TAG=1 PUSH_INNER_TAG=1	Don't care	ES0 tag A is pushed as outer tag. ES0 tag B is pushed as inner tag.
PUSH_OUTER_TAG=2 PUSH_INNER_TAG=1	Don't care	Port tag is pushed as outer tag. This overrules port settings in TAG_CFG. ES0 tag B is pushed as inner tag.
PUSH_OUTER_TAG=3 PUSH_INNER_TAG=1	Don't care	No outer tag is pushed. This overrules port settings in TAG_CFG. ES0 tag B is pushed as inner tag. ES0 tag B is effectively the outer tag, because no outer tag is pushed.

4.16.1.4 Constructing VLAN Tags

When pushing a VLAN tag, the contents of the tag header, including the TPID, is highly programmable. The starting point is the classified tag header coming from the analyzer containing the PCP, DEI, VID and tag type. For each of the fields in the resulting tag, it is programmable how the value is determined. For more information, see Figure 4-30.

FIGURE 4-30: TAG CONSTRUCTION (PORT TAG, ES0 TAG A, ES0 TAG B)



The port tag, ES0 tag A, and ES0 tag B have individual configurations. For the port tag, the available tag field options are:

Port tag: PCP

- Use the classified PCP.
- Use the egress port's port VLAN (PORT_VLAN.PORT_PCP).
- Map the DP level and QoS class to a new PCP value using the per-port table PCP_DEI_QOS_MAP_CFG.
- Use the QoS class directly.

Port tag: DEI

- Use the classified DEI.
- Use the egress port's port VLAN (PORT_VLAN.PORT_DEI).
- Map the DP level and QoS class to a new DEI value using the per-port table PCP_DEI_QOS_MAP_CFG.
- Use the DP level directly.

Port tag: VID

- Use the classified VID.
- Use the egress port's port VLAN (PORT_VLAN.PORT_VID).

Port tag: TPID

- Use Ethernet type 0x8100 (C-tag).
- Use Ethernet type 0x88A8 (S-tag).
- Use custom Ethernet type programmed in PORT_VLAN.PORT_TPID.
- Use custom Ethernet type programmed in PORT_VLAN.PORT_TPID unless the incoming tag was a C-tag in which case Ethernet type 0x8100 is used.

Similar options for the ES0 tag A and ES0 tag B are available:

ES0 tag: PCP

- Use the classified PCP.
- Use ES0_ACTION.PCP_A_VAL for ES0 tag A and use ES0_ACTION.PCP_B_VAL for ES0 tag B.
- Map the DP level and QoS class to a new PCP using the per-port table PCP_DEI_QOS_MAP_CFG.
- Use the QoS class directly.

ES0 tag: DEI

- Use the classified DEI.
- Use ES0_ACTION.DEI_A_VAL for ES0 tag A and use ES0_ACTION.DEI_B_VAL for ES0 tag B.
- Map the DP level and QoS class to a new DEI using the per-port table PCP_DEI_QOS_MAP_CFG.
- Use the DP level directly.

ES0 tag: VID

- Use the classified VID incremented with ES0_ACTION.VID_A_VAL for ES0 tag A and use the classified VID incremented with ES0_ACTION.VID_B_VAL for ES0 tag B.
- Use ES0_ACTION.VID_A_VAL for ES0 tag A and use ES0_ACTION.VID_B_VAL for ES0 tag B.

ES0 tag: TPID

- Use Ethernet type 0x8100 (C-tag).
- Use Ethernet type 0x88A8 (S-tag).
- Use custom Ethernet type programmed in PORT_VLAN.PORT_TPID.
- Use custom Ethernet type programmed in PORT_VLAN.PORT_TPID unless the incoming tag was a C-tag in which case Ethernet type 0x8100 is used.

4.16.2 VLAN PUSHING

In the final VLAN editing step, the VLAN tags derived from the previous steps are pushed to the frame.

4.16.3 REDUNDANCY TAG EDITING

Table 4-38 lists the possible combinations of R-TAG actions depending on the push/pop indication from the stream table in the analyzer, the frame's current R-TAG condition, and the RED_TAG_CFG configuration for the egress port.

The combinations shown below determine R-TAG editing for known streams (STREAM_ID = 1) as STREAM_ACTION is only valid for known streams. For unknown streams (STREAM_ID = 0), the R-TAG action is always 'No operation'.

TABLE 4-38: R-TAG EDITING COMBINATIONS

R-TAG operation from analyzer	Frame already contains an R-TAG?	RED_TAG_CFG	R-TAG action
No operation	No	Don't care	No operation
Push R-TAG	No	Disable	No operation
Push R-TAG	No	Enable	Push R-TAG
No operation or pop R-TAG	Yes	Disable	Pop R-TAG
No operation	Yes	Enable	No operation (for example, only splitting)

TABLE 4-38: R-TAG EDITING COMBINATIONS (CONTINUED)

R-TAG operation from analyzer	Frame already contains an R-TAG?	RED_TAG_CFG	R-TAG action
Pop R-TAG	Yes	Enable	Pop R-TAG

Note: The analyzer ensures that invalid combinations are not possible. For instance, it is not possible to push an R-TAG for an already R-TAGGED frame and it is not possible to pop an R-TAG from an untagged frame. Finally, is not possible to pop and push an R-TAG for the same frame.

R-TAG editing is done independent of VLAN editing. However, in case of an R-TAG push, at most one more VLAN tag can be pushed. The R-TAG is always pushed after the VLAN tag so that the R-TAG becomes the innermost tag of tags pushed by the rewriter. The combination of pushing 2 VLAN tags along with an R-TAG is not supported. A sticky bit ES0_TAGB_PUSH_FAILED is set to indicate this error.

4.16.4 DSCP REMARKING

The rewriter can remark the DSCP value in IPv4 and IPv6 frames, that is, write a new DSCP value to the DSCP field in the frame.

If a port is enabled for DSCP remarking (DSCP_CFG.DSCP_REWR_CFG), the new DSCP value is derived by using the classified DSCP value from the analyzer (the basic classification or the VCAP IS1) in the ingress port. This DSCP value can be mapped before replacing the existing value in the frame. The following options are available:

- No DSCP remarking - Leave the DSCP value in the frame untouched.
- Update the DSCP value in the frame with the value received from the analyzer
- Update the DSCP value in the frame with the value received from the analyzer remapped through DSCP_REMAP_CFG. This is done independently of the value of the drop precedence level.
- Update the DSCP value in the frame with the value received from the analyzer remapped through DSCP_REMAP_CFG or DSCP_REMAP_DP1_CFG dependent on the drop precedence level. This enables one mapping for green frames and another for yellow frames so that the resulting DSCP value can reflect the color of the frame.

In addition, the IP checksum is updated for IPv4 frames. Note that the IPv6 header does not contain a checksum. As a result, checksum updating does not apply for IPv6 frames.

4.16.5 FCS UPDATING

The rewriter updates a frame's FCS when required or instructed to do so. Different handling is available for frames injected by the CPU and for all other frames.

For non-CPU injected frames, the following update options are available:

- Never update the FCS.
- Conditional update: Update the FCS if the frame was modified by the rewriter, for instance due to PTP time stamping, VLAN tagging, or DSCP remarking.
- Always update the FCS.

In addition, the rewriter can update the FCS for all frames injected from the CPU through the CPU injection queues in the CPU port module:

- Never update the FCS.
- Always update the FCS.

4.16.6 ETHERCAT TIME STAMPING

The device supports time stamping of EtherCAT frames (Ethertype=0x88A4). This is signaled through VCAP IS2 action REW_OP[4:0] = 4. The EtherCAT frames are handled in rewriter similar to two-step PTP frames. Both the EtherCAT frames and the PTP two-step frames share the same egress Tx FIFO to store the TX time stamps.

4.16.7 MAC ADDRESS REWRITER OPERATIONS

The rewriter supports two operations on a frame's MAC addresses, triggered through VCAP IS2 action REW_OP.

The first operation swaps the frame's MAC addresses:

- The original destination MAC address becomes the new source MAC address.

- The original source MAC address becomes the new destination MAC address.

Bit 40 is cleared in the new source MAC address to prevent the possibility of transmitting an illegal frame with a multicast source MAC address.

The swapping of MAC addresses is useful when implementing a general hairpinning functionality where an IS2-identified flow is hardware looped back to the source port while swapping the MAC addresses.

The second operation replaces the frame's source MAC address. The new source MAC address is configurable per egress port through REW:PORT:PTP_SMAC_LOW and REW:PORT:PTP_SMAC_LOW.

The operations of swapping of MAC addresses and replacing the source MAC address are mutually exclusive and must not be enabled at the same time.

4.17 PRP/HSR

The device supports DANP functionality for PRP and DANH functionality for HSR, with tracking of sequence numbers and discarding duplicates. Only one DANP or DANH instance can be supported at the same time. FRER is not supported at the same time on ports that are used as LRE ports.

4.17.1 HSR DANH OVERVIEW

LAN9645xF supports two LRE ports (LREA and LREB) towards the HSR ring and a link layer interface towards either the internal CPU or an external CPU located on the NPI. Any two front ports (ports 0-8) can be assigned as LRE ports. As a link layer interface, the internal CPU ports or an external NPI can be used. The CPU implements the DANH functionality with hardware-assisted offloads for operations such as sequence number generation, duplicate discard, and frame forwarding.

The frame forwarding logic uses existing handles such as stream table, MAC table, PGID table, source masks, VLAN table, and VCAP S2 actions. Initially, a port-based VLAN can be set up that includes the CPU and the two LRE ports.

Frames injected by the DANH into the switch core should not be forwarded when received again on an LRE port. This can be prevented by installing a VCAP S2 entry matching against a source MAC address and ingress port.

LAN9645xF can generate sequence numbers for frames injected by the DANH. This functionality is identical to FRER sequence number generation and requires the following configuration:

- Frames injected from CPU must have IFH.BYPASS set to 0.
- Classify frames moving to an ISDX using VCAP S1.
- Set up the stream table for the ISDX for sequence number generation with SEQ_GEN_ENA set to 1.
- The stream table split mask (ANA::SPLIT_MASK) can be set to include both LRE ports to ensure forwarding from the DANH to both LRE ports.

The device does not support a dedicated HSR nodes table, but uses ISDX classification and the stream table to track known hosts. For each known host, the following configuration is required:

- Assign two ISDXs per host and install two VCAP S1 entries with ISDX classification – one entry for frames from the host received on LREA and one entry for frames from the host received on LREB.
- Ingress ISDX counters track the number of frames from the host per LRE.
- TimeLastSeen is tracked through the stream table (ANA::STREAM_TIME). The time used has a configurable period, typically 100ms.

The classifier supports extraction to the CPU of the following DANH-related frames:

- Supervision frames with EtherType = 0x88FB
- Frames with no HSR tag

The analyzer supports a set of statistics for the LREA and LREB ports counting number of frames with HSR tags.

4.17.2 PRP DANP FUNCTIONALITY OVERVIEW

LAN9645xF supports two LRE ports (LREA and LREB) towards the PRP network and a link layer interface towards either the internal CPU or an external CPU located on the NPI. Any two front ports (port 0-8) can be assigned as LRE ports. As data link interface, the internal CPU ports or an external NPI can be used. The CPU implements the DANP functionality with hardware assisted offloads for operations such as sequence number generation, duplicate discard, and frame forwarding.

The frame forwarding logic uses existing handles such as stream table, MAC table, PGID table, source masks, VLAN table, and VCAP S2 actions.

LAN9645xF generates sequence numbers for frames injected by the DANP. This functionality is identical to FRER sequence number generation and requires the following configuration:

- Frames injected from CPU must have IFH.BYPASS set to 0.
- Classify frames to an ISDX using VCAP S1.
- Set up the stream table for the ISDX for sequence number generation with SEQ_GEN_ENA set to 1.
- The stream table split mask (ANA::SPLIT_MASK) can be set to include both LRE ports to ensure forwarding from the DANP to both LRE ports.

LAN9645xF does not support a dedicated PRP nodes table, but uses ISDX classification and the stream table to track known hosts. For each known host, the following configuration is required:

- Assign two ISDXs per host and install two VCAP S1 entries with ISDX classification – one entry for frames from the host received on LREA and one entry for frames from the host received on LREB.
- Ingress ISDX counters track the number of frames from the host per LRE.
- TimeLastSeen is tracked through the stream table (ANA::STREAM_TIME). The time used has a configurable period, typically 100ms.

The classifier supports extraction to the CPU of the following DANP-related frames:

- Supervision frames with EtherType = 0x88FB

The analyzer supports a set of statistics for the LREA and LREB ports, counting number of frames with RCT trailers.

LAN9645xF recognizes RCT trailers in incoming frames and uses this information for processing. RCT trailers are not removed from frames when forwarding to the CPU and must be handled by the CPU if required.

LAN9645xF does not push RCT trailers on frames injected by the CPU, but assists the CPU by updating relevant information. Prior to injection, the CPU inserts the 6-byte RCT trailer in the frame and pre-formats the trailer with a PRP suffix and frame size. When passing through the switch, LAN9645xF rewrites the sequence number and LAN identifier in the RCT trailer based on the frame processing.

4.17.3 FRAME PROCESSING

4.17.3.1 Identifying HSR Tags

Each LRE ingress port must be enabled for identifying and using incoming HSR tags. LAN9645xF identifies HSR tags, when located, as outer tag or as inner tag after one VLAN tag.

Note: A port cannot look for both FRER RTAGs and HSR tags at the same time.

When identifying an HSR tag, the ingress frame is modified to align with a 32-bit word boundary by stripping the HSR EtherType from the frame. This modification is signaled in the IFH by setting IFH.RTAGD to:

- 1, if the HSR tag is an outer tag
- 2, if the HSR tag is an inner tag
- 0, means that no HSR tag was found and that the frame is unmodified

When transmitting the frame again on an egress front port, changes to the frame are reverted by inserting the HSR EtherType again. The frame is not rewritten towards the CPU; this means that the removal of the HSR EtherType is not reverted and the CPU receives a frame where the HSR EtherType is missing. The IFH.RTAGD > 0 signals that this is the status.

The HSR tag's LSDU size is not checked against the frame's actual size. Frames of any LSDU size are accepted.

4.17.3.2 Identifying RCT Trailers

RCT classification is done at end-of-frame when the frame's potential RCT data is available as the 6 bytes immediately before the frame's FCS. The RCT classification determines whether a frame carries an RCT and extracts the following set of RCT parameters from the trailer:

- PRP Suffix
- Lan Identifier (LanId).
- Sequence number (SeqNr)
- Frame size (LSDUsize)

RCT classification is enabled in ANA:PORT:RED_CFG.PRP_AWARE_ENA. To accept the incoming frame data as an RCT, the following conditions must be met:

- The RCT's PRP suffix must equal 0x88FB.
- The RCT's LSDUsize must equal the frame's actual LSDU size. Up to one VLAN tag is subtracted from the frame's LSDU size.
- The RCT's LanId must equal the port configuration (ANA:PORT:RED_CFG.NETID and ANA:PORT:RED_CFG.LANID). The port's LanId is calculated as (NETID<<1)+LANID and compared against the RCT.

The LanId check can optionally be disabled through ANA:PORT:RED_CFG.ACCEPT_LANID_ERR_ENA. Frames failing the LanId check are counted in ANA::CNT_RX_WRONG_LAN_A or ANA::CNT_RX_WRONG_LAN_B.

4.17.3.3 Classification

The classifier provides handles to extract specific frames in relation to HSR and PRP:

- Supervision frames with EtherType = 0x88FB can be copied to the CPU.
- Frames with no HSR tags can be redirected to the CPU.

4.17.3.4 PRP Processing

PRP processes specific forwarding decisions, which is enabled through PRP_ENA.

If a frame triggers a port move in the MAC address table from LREA to LREB (or the opposite), the associated MAC table entry is updated to use destination PGID_SAN_A_B, and the entry locked in the MAC table. The sticky bit SAN_A_B_RELEARN is set.

PGID mask PGID_SAN_A_B contains both LRE ports in its mask. This ensures that frames sent to the destination are sent to both ports.

Setting the SAN_ENA flag for a PGID entry ensures that MAC table entries using the PGID are handled as SAN destinations; sequence numbers are not generated and any RCT rewrites in the rewriter are disabled.

4.17.3.5 Stream Table

The stream table is used with different purposes for DANP and DANH functionality. The stream table provides the sequence number generator which is used the same way as FRER sequence number generation. The sequence number is carried to the egress in the IFH.SEQ_NUM.

The stream table also provides Nodes Table functionality by tracking TimeLastSeen and the associated counters per known host. It is the responsibility of the CPU to install hosts in the stream table upon receiving supervision frames.

Each time a frame is seen from a known host, the associated entry in the stream table latches the time. This is done by sampling the value of the current time from ANA::TIME_CFG.CURRENT_TIME. The current time increments with a preconfigured period (ANA::TIME_CFG.PRESCALER) and wraps at approximately 4095. Typically, a period of 10-100ms is sufficient granularity and allows software to read the stream table entries approximately every few seconds.

ISDX statistics provide counters that allow software to track the number of frames received on a given port from a known host.

4.17.4 DUPLICATE DISCARD

The duplicate discard functionality is a part of the queue system. It operates on each egress frame copy made from an ingress frame based on the forwarding decision made in the analyzer. Duplicate discard can be enabled for up to three ports (LREA, LREB, CPU) selected from any of the front and CPU ports. LREA, LREB, and CPU are denoted as DD ports in this section.

A frame undergoes a duplicate discard check if the IFH.DUPL_DISC_ENA is enabled for that frame. In the case of PRP, this setting is enabled if the frame is received with a valid RCT on a port configured for PRP. For HSR, it is enabled if the frame is received with an HSR tag on a port configured for HSR, or if a sequence number is generated for the frame. However, you can disable duplicate discard for a frame that would otherwise be subject to it by using the VCAP S2 action DUPL_DISC_DIS.

To enable basic duplicate discard operation, the following settings must be applied:

- QSYS::MISC_DROP_CFG.FRER_ENA = 1.
- QSYS::DD_CFG.CPU_PORT = <CPU port number or NPI port number>.
- QSYS::DD_CFG.LREA_PORT = <port number>.
- QSYS::DD_CFG.LREB_PORT = <port number>.
- QSYS::DD_CFG.DUPL_DISC_MASK = 0x7 (DANH) or 0x4 (DANP).

- QSYS::DD_CFG.UPD_DUPL_DISC_MASK = 0x7 (DANH) or 0x4 (DANP).

4.17.4.1 Duplicate Discard Table

The duplicate discard table contains information about hosts and their sequence numbers observed on an egress port. The duplicate discard table discards a frame when the same sequence number from a host is seen more than once on an egress port.

The table is organized as a hash table with eight buckets in each row. Each row is indexed based on a hash value calculated from the host's source MAC address and sequence number. The duplicate discard table is looked up once per frame per egress port.

The Duplicate Discard table is accessed through registers in QSYS:DD_DISC_TBL.

- Entries in the duplicate discard table are added, deleted, or updated by the following methods:
- Automatic hardware-based insertion or update of an entry based on transmission of a frame on an egress port.
- CPU access to the duplicate discard table – for example CPU-based table maintenance.
- Automated age scans.

The possible types of access are listed in Table 4-39.

TABLE 4-39: CPU ACCESS TO DUPLICATE DISCARD TABLE

Command	Purpose	Use
Learn	Insert/learn new entry in duplicate discard table. Position is given by HASH(MAC).	First, the CPU provides the command 0x0 in the CPU_ACCESS_CMD field of the DISC_ACCESS_CTRL register. It then provides the duplicate discard entry data by writing to the respective registers: • DISC_ACCESS_CFG_0 • DISC_ACCESS_CFG_1 • DISC_ACCESS_CFG_2 • DISC_ACCESS_STAT_0 • DISC_ACCESS_STAT_1 • DISC_ACCESS_STAT_2 • DISC_ACCESS_STAT_3 Finally, it starts the learning process by writing 0x1 in the DISC_TABLE_ACCESS_SHOT field of the register. The process is complete when the CPU reads that the value is 0x0.
Unlearn/Forget	Delete/unlearn entry in duplicate discard table. Position is given by HASH(MAC).	First, the CPU provides the command 0x1 in the CPU_ACCESS_CMD field of the DISC_ACCESS_CTRL register. It then provides the MAC in two parts: the MSB 16-bit in the field DISC_ENTRY_MAC_MSB and the LSB 32-bit in the field DISC_ENTRY_MAC_LSB. Finally, it starts the unlearning process by writing 0x1 in the DISC_TABLE_ACCESS_SHOT field of the register DISC_ACCESS_CTRL. The process is complete when the CPU reads back the value 0x0.

TABLE 4-39: CPU ACCESS TO DUPLICATE DISCARD TABLE (CONTINUED)

Command	Purpose	Use
Lookup	Lookup entry in duplicate discard table. Position is given by HASH(MAC).	First, the CPU provides the command 0x2 in the CPU_ACCESS_CMD field of the DISC_-ACCESS_CTRL register. It then provides the MAC in two parts: the MSB 16-bit in the field DISC_EN-TRY_MAC_MSB and the LSB 32-bit in the field DISC_ENTRY_MAC_LSB. Finally, it starts the lookup process by writing 0x1 in the DISC_TABLE_ACCESS_SHOT field of the register DISC_ACCESS_CTRL. The process is complete when the CPU reads back the value 0x0.
Read direct	Read entry in duplicate discard table indexed by (row, column).	First, the CPU provides the command 0x3 in the CPU_ACCESS_CMD field of the DISC_-ACCESS_CTRL register. It then provides the starting row and column by writing to the CPU_ACCESS_DI-RECT_ROW and CPU_ACCESS_DIRECT_COL fields, respectively. Finally, it starts the entry reading process of the table from the specified (row, column) by writing 0x1 in the DISC_TABLE_ACCESS_-SHOT (0) field of the register. The process is complete when the CPU reads the value is 0x0.
Write direct	Write entry in duplicate discard table indexed by (row, column)	First, the CPU provides the command 0x4 in the CPU_ACCESS_CMD field of the DISC_-ACCESS_CTRL register. It then provides the starting row and column by writing to the fields CPU_ACCESS_DI-RECT_ROW and CPU_ACCESS_DI-RECT_COL fields respectively. It proceeds to write the duplicate discard entry data by writ-ing to the respective registers: • DISC_ACCESS_CFG_0 • DISC_ACCESS_CFG_1 • DISC_ACCESS_CFG_2 • DISC_ACCESS_STAT_0 • DISC_ACCESS_STAT_1 • DISC_ACCESS_STAT_2 • DISC_ACCESS_STAT_3 Finally, it starts the entry writing process of the table from the specified (row, column) by writing 0x1 in the DISC_TABLE_ACCESS_-SHOT (0) field of the register. The process is complete when the CPU reads the value is 0x0.

TABLE 4-39: CPU ACCESS TO DUPLICATE DISCARD TABLE (CONTINUED)

Command	Purpose	Use
Clear all	Initialize the table.	First, the CPU provides the command 0x0 in the CPU_ACCESS_CMD field of the DISC_-ACCESS_CTRL register. It then provides the starting row by writing to the CPU_ACCESS_DIRECT_ROW field of the register. Finally, it starts the automated clearing process of the table from the specified row by writing 0x1 in the DISC_TABLE_ACCESS_-SHOT (0) field of the register. The process is complete when the CPU reads the value is 0x0.

4.17.4.2 Automated Aging

Automated aging implements a periodic process that updates the age fields of all the entries in a row. In every periodic cycle, the process reads out a row and increments the fields DISC_ENTRY_AGE_FLAG_0, DISC_ENTRY_AGE_-FLAG_1 and DISC_ENTRY_AGE_FLAG_2. If any of the fields reaches the value 0x7 (meaning that it aged seven times) then the duplicate discard entry is deleted/disabled.

The automated aging process can be controlled in two ways, periodically or triggered manually. The duplicate discard table currently has only one aging interval available. Automated aging can be enabled using the field AUTOAGE_IN-TERVAL_ENA. The CPU writes the period per row in field PERIOD_VAL in UNIT_SIZE amount of clock cycles. For example, if PERIOD_VAL = 1, the hardware will age one row every UNIT_SIZE clock cycle. The UNIT_SIZE values are encoded to allow for different levels of granularity. The actual value written in the UNIT_SIZE field is shifted by the hardware to produce the following encoding: 0=8 clock cycles, 1=128, 2=2048 and 3=32768.

Additionally, the aging process can be triggered manually using the FORCE_HW_SCAN_SHOT field. This triggers an instant hardware autoage scan (once any on-going scan is completed) starting from the row specified in NEX-T_AGED_ROW. The bit FORCE_HW_SCAN_SHOT is cleared by hardware when a row scan completes and the NEX-T_AGED_ROW field is incremented.

4.17.5 REWRITING

The device can pop or push one HSR tag. Whether to pop or push an HSR tag follows the same decision making as the FRER RTAGs logic. To push an HSR tag (RED_TAG_CFG=1), the egress port must be enabled for redundancy tagging. Any outer VLAN tag must be popped before a HSR tag can be popped.

When pushing an HSR tag, the rewriter constructs the HSR tag based on information from the IFH and egress port configurations.

If the rewriter pushes both a VLAN tag and an HSR tag, the HSR tag is always the inner tag.

The rewriter does not rewrite the frame towards the CPU. If the frame contains an HSR tag, this is signaled in the frame's IFH through IFH.RTAGD > 0.

The device can rewrite information in RCT trailers in outgoing frames. The decision to rewrite the RCT trailer follows the same logic as for FRER RTAGs. The egress port must be enabled for PRP by setting SYS::PORT_MODE.PRP_ENA. The device does not insert an RCT into the frame – it assumes that the injecting CPU has already inserted a preformatted RCT, which is then updated by the rewriter.

When rewriting the RCT, the resulting RCT trailer is based on already existing values in the frame, as well as information from the IFH and egress port configuration.

The rewriter assumes the RCT is located immediately before the FCS. There is no check to validate this.

4.17.6 STATISTICS AND DIAGNOSTICS

The analyzer contains a set of counters that can inform about the frame processing.

The duplicate discard table contains a set of counters and sticky bits that can inform about the frame processing.

4.18 CPU Port Module

The CPU port module (DEVCPU) contains eight CPU extraction queues and two CPU injection queues. These queues provide an interface for exchanging frames between an external CPU system and the switch. In addition, any Ethernet interface on the device can be used for extracting and injecting frames. The Ethernet interface used in this way is called a node processor interface (NPI).

Injected frames may be prepended with an injection header to control processing and forwarding of these frames. Extracted frames may be prepended with an extraction header to supply frame arrival and classification information associated with each frame.

4.18.1 FRAME EXTRACTION

In the switch core, CPU extracted frames are forwarded to one of the eight CPU extraction queues. Each of these queues is mapped to one of two CPU ports (port 9 and port 10) through QSYS::CPU_GROUP_MAP. For each CPU port, there is a scheduler working either in strict mode or round robin, which selects between the CPU extraction queues mapped to the same CPU port. In strict mode, higher queue numbers are preferred over smaller queue numbers. In round robin, all queue are serviced one after another.

The two CPU ports contain the same rewriter as regular front ports. The rewriter modifies the frames before sending them to the CPU. In particular, the rewriter inserts an extraction header (SYS::PORT_MODE.INCL_XTR_HDR), which contains relevant side band information about the frame such as the frame's classification result (VLAN tag information, DSCP, QoS class), ingress time stamp, and the reason for sending the frame to the CPU. For more information about the rewriter, see Section 4.11.4, Forwarding Engine.

Frames are read out through register access (DEVCPU_QS:XTR).

Table 4-40 lists the contents of the CPU extraction header.

TABLE 4-40: CPU EXTRACTION HEADER

Field	Bit	Width	Description
TIMESTAMP	186	38	Frame's ingress time stamp.
RESERVED	185	1	Field is not used by extraction.
LEN	171	14	Frame's length.
WRDMODE	169	2	Internal cell filling mode.
RTAGD	167	2	Mask indicating redundancy tag location. Bit 0: Outer tag Bit 1: Inner tag
CUTTHRU	166	1	If set, frame was cut-through forwarded.
REW_CMD	156	10	Rewriter command, bits 9:0.
REW_OAM	155	1	If set, the frame is subject to OAM-related rewriting depending on the OAM PDU type. PDU_TYPE encodes the OAM PDU type.
PDU_TYPE	151	4	PDU type: 0: NONE 1: Y1731_CCM 2: MRP_TST 3: MRP_ITST 4: DLR_BCN 5: DLR_ADV 6: RESERVED 7: IPV4 8: IPV6 9: Y1731_NON_CCM
FCS_UPD	150	1	If set, the frame's FCS is forced to be updated before transmission.
DP	149	1	Drop precedence (DP) level after policing.
RESERVED	148	1	Field is not used by extraction.
POP_CNT	146	2	Number of tags to pop from frame.
ETYPE_OFS	144	2	Number of tags located in front of the EtherType.
SRCPORT	140	4	Logical source port of frame.

TABLE 4-40: CPU EXTRACTION HEADER (CONTINUED)

RESERVED	136	4	Field is not used by extraction.
SEQ_NUM	120	16	IEEE 802.1CB sequence number, either extracted from frame's R-TAG or generated by analyzer.
TCI	103	17	Classified TCI information: Bit 16: The tag information's associated Tag Protocol Identifier (TPID). The definitions are: 0: C-tag: EtherType = 0x8100 1: S-tag: EtherType = 0x88A8 or custom value Bits 15-13: Classified PCP Bit 12: Classified DEI Bits 11-0: Classified VID
DSCP	97	6	Classified DSCP value. For gPTP frames, this field contains rewriter command, bits 15:10.
QOS_CLASS	94	3	Classified QoS class.
CPUQ	86	8	CPU extraction queue mask (one bit per CPU extraction queue). Each bit set implies the frame was subjected to CPU forwarding to the specific queue.
LEARN_FLAGS	84	2	The source MAC address learning action triggered by the frame. 0: No learning 1: Learning a new entry 2: Updating an already learned unlocked entry 3: Updating an already learned locked entry
SFLOW_ID	80	4	sFlow sampling ID. 0-9: Frame was sFlow sampled by a Tx sampler on port given by SFLOW_ID 10: Frame was sFlow sampled by an Rx sampler on port given by SRCPORT 15: Frame was not sFlow sampled
ACL_HIT	79	1	Super priority: Set ACL_HIT to 0 Set ACL_IDX[4] to 1
ACL_IDX	73	6	See ACL_HIT.
ISDX	65	8	Classified ISDX.
DSTS	55	10	Destination port set.
FLOOD	53	2	Applied storm policer: 0: None 1: Unicast 2: Multicast 3: Broadcast
SEQ_OP	51	2	Egress R-TAG operation: 0: None 1: Reserved 2: Push R-TAG 3: Pop R-TAG
IPV	48	3	Internal priority value. Used for queuing.
RESERVED	11	37	Field is not used by extraction.
DUPL_DISC_ENA	10	1	If set, the frame is not subject to duplicate discard at egress port.
RCT_AVAIL	9	1	If set, frame contains an RCT.
RESERVED	0	9	Field is not used by extraction.

4.18.2 FRAME INJECTION

Frames are injected through register accesses (DEVCPU_QS:INJ) to one of the two CPU injection groups. The injection groups connect to the two CPU ports (port 9 and port 10) in the CPU port module. In the CPU port module, each of the two CPU ports have dedicated access to the switch core. Inside the switch core, all CPU injected frames are seen as coming from CPU port 9. This implies that both CPU injection groups consume memory resources from the shared queue system for port 9 and that analyzer configuration for port 9 are applied to all frames.

In the switch core, the CPU ports can be preferred over other ingress ports when transferring frames to egress queues by enabling precedence of the CPU ports (QSYS::EQ_PREFER_SRC).

The first 28 bytes of a frame written to a CPU injection group is an injection header containing relevant side band information about how the frame must be processed by the switch core. The CPU ports must be enabled to expect the CPU injection header (SYS::PORT_MODE.INCL_INJ_HDR).

On a per-frame basis, the CPU controls whether frames injected through the CPU port module are processed by the analyzer. If the frame is processed by the analyzer, it is sent through the processing steps to calculate the destination ports for the frame. If analyzer processing is not selected, the CPU can specify the destination port set and related information to fully control the forwarding of the frame. For more information about the analyzer's processing steps, see Section 4.11.4, Forwarding Engine.

Table 4-41 lists the contents of the CPU extraction header.

TABLE 4-41: CPU INJECTION HEADER

Field	Bit	Width	Description
TIMESTAMP	192	32	Frame's ingress time stamp (least 32 bits).
BYPASS	191	1	When this bit is set, the analyzer processing is skipped for this frame. Relevant information is extracted from the prefilled IFH. In particular, the destination set is specified in DSTS and CPUQ. When this bit is cleared, the frame is subject to normal frame processing by the analyzer including lookups in the MAC table and VLAN table.
MASQ	190	1	Masqueraded injection with MASQ_PORT defining the frame's logical source port.
MASQ_PORT	186	4	Masqueraded port number for injection.
RCT_INJ	185	1	If set, frame contains preformatted RCT trailer for PRP processing.
RESERVED	171	14	Field is not used by injection. Must be set to 0.
RESERVED	169	2	Field is not used by injection. Must be set to 0.
RTAGD	167	2	Mask indicating redundancy tag location. Bit 0: Outer tag Bit 1: Inner tag
RESERVED	166	1	Field is not used by injection. Must be set to 0.
REW_CMD	156	10	Rewriter command, bits 9:0.
REW_OAM	155	1	If set, the frame is subject to OAM-related rewriting depending on the OAM PDU type. PDU_TYPE encodes the OAM PDU type.
PDU_TYPE	151	4	PDU type: 0: NONE 1: Y1731_CCM 2: MRP_TST 3: MRP_ITST 4: DLR_BCN 5: DLR_ADV 6: RESERVED 7: IPV4 8: IPV6 9: Y1731_NON_CCM
FCS_UPD	150	1	If set, the frame's FCS is forced to be updated before transmission.

TABLE 4-41: CPU INJECTION HEADER (CONTINUED)

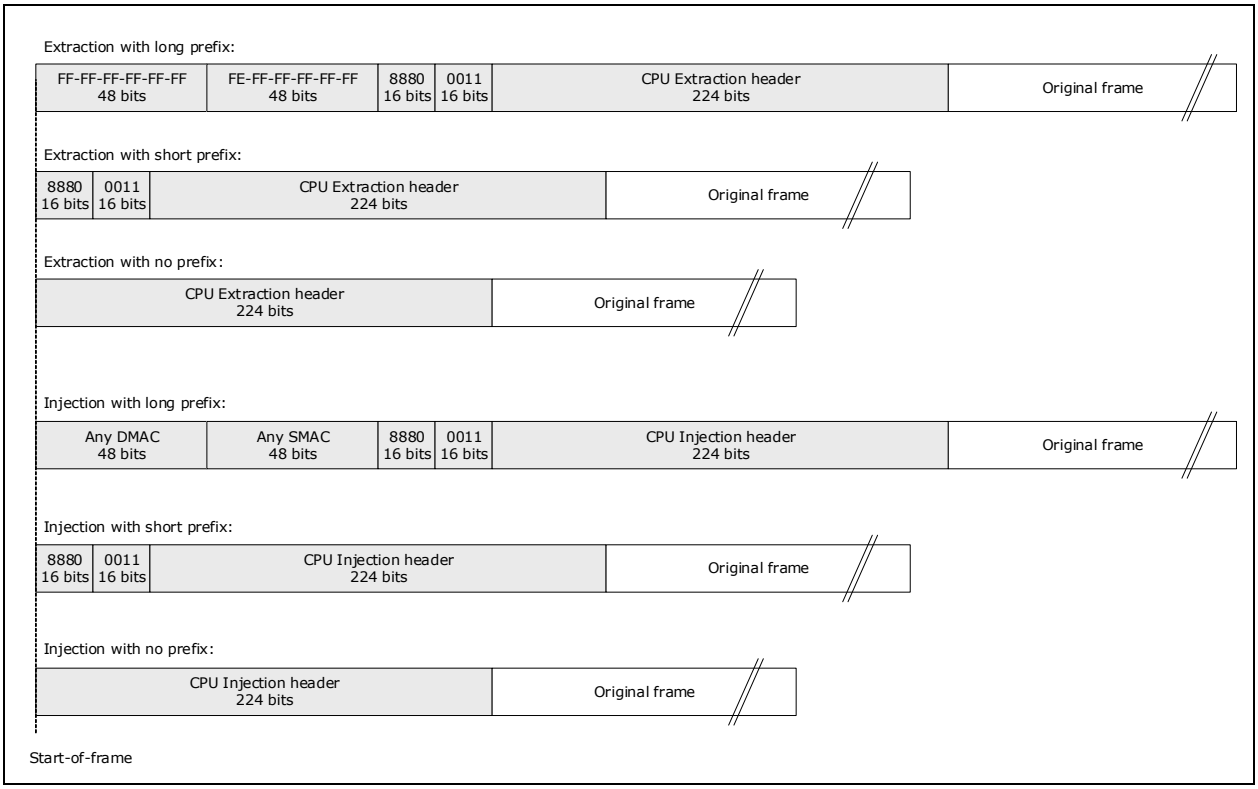
DP	149	1	Drop precedence (DP) level after policing.
RESERVED	148	1	Field is not used by injection. Must be set to 0.
POP_CNT	146	2	Number of tags to pop from frame.
ETYPE_OFS	144	2	Number of tags located in front of the EtherType.
RESERVED	136	8	Field is not used by injection. Must be set to 0.
SEQ_NUM	120	16	IEEE 802.1CB sequence number, either extracted from frame's R-TAG or generated by analyzer
TCI	103	17	Classified TCI information: Bit 16: The tag information's associated Tag Protocol Identifier (TPID). The definitions are: 0: C-tag: EtherType = 0x8100 1: S-tag: EtherType = 0x88A8 or custom value Bits 15-13: Classified PCP Bit 12: Classified DEI Bits 11-0: Classified VID
DSCP	97	6	Classified DSCP value. For gPTP frames, this field contains rewriter command, bits 15:10.
QOS_CLASS	94	3	Classified QoS class.
CPUQ	86	8	CPU extraction queue mask (one bit per CPU extraction queue). Each bit set implies the frame was subjected to CPU forwarding to the specific queue.
RESERVED	84	2	Field is not used by injection. Must be set to 0.
SFLOW_ID	80	4	sFlow sampling ID. 0-9: Frame was sFlow sampled by a Tx sampler on port given by SFLOW_ID 10: Frame was sFlow sampled by an Rx sampler on port given by SRCPORT 15: Frame was not sFlow sampled
ACL_HIT	79	1	Super priority: Set ACL_HIT to 0 Set ACL_IDX[4] to 1
ACL_IDX	73	6	See ACL_HIT.
ISDX	65	8	Classified ISDX.
DSTS	55	10	Destination port set.
RESERVED	53	2	Field is not used by injection. Must be set to 0.
SEQ_OP	51	2	Egress R-TAG operation: 0: None 1: Reserved 2: Push R-TAG 3: Pop R-TAG
IPV	48	3	Internal priority value. Used for queuing.
AFI	47	1	Injected into AFI.
RESERVED	11	36	Field is not used by injection. Must be set to 0.
DUPL_DISC_ENA	10	1	If set, the frame is not subject to duplicate discard at egress port.
RESERVED	0	10	Field is not used by injection. Must be set to 0.

4.18.3 NODE PROCESSOR INTERFACE (NPI)

Any front port can be configured as an NPI through which frames can be injected from and extracted to an external CPU. Only one port can be an NPI at the same time. QSYS::EXT_CPU_CFG.EXT_CPU_PORT holds the port number of the NPI.

Frames being extracted by the external CPU can have the CPU extraction header inserted in front of the frame (SYS::PORT_MODE.INCL_XTR_HDR). Similarly, frames being injected into the switch core by the external CPU can have the CPU injection header inserted in front of the frame (SYS::PORT_MODE.INCL_INJ_HDR). In addition, there three different options in terms of inserting a prefix in front of the CPU injection or extraction header. Figure 4-31 shows the different frame formats supported.

FIGURE 4-31: CPU INJECTION AND EXTRACTION PREFIXES



Note that inserting a CPU extraction header in front of a frame disables all other frame modifications done in the rewriter.

When injecting frames with an CPU injection header all incoming frames are expected to adhere to the configured prefix mode. The following parsing of the frames takes place:

- No prefix. All incoming frames are parsed as if they have a CPU injection header in front of the frame. Forwarding is based on the instructions in the CPU injection header.
- Short prefix. Incoming frames are checked against the expected format (start of frame must include 0x88800011). If the prefix does not match then an error indication is set (SYS::PORT_MODE.INJ_HDR_ERR). Compliant frames are forwarded based on the instructions in the CPU injection header. Non-compliant frames are forwarded as normal frames where the prefix and CPU injection header equaling the first 32 bytes of the frame are skipped.
- Long prefix. All incoming frames are parsed as if they have a long prefix followed by the CPU injection header in front of the frame. The incoming frames are checked against the expected format (start of frame must include 12 bytes of MAC addresses followed by 0x88800011). If the prefix does not match then an error indication is set (SYS::PORT_MODE.INJ_HDR_ERR). Both compliant and non-compliant frames are forwarded based on the instructions in the CPU injection header.

The external CPU can control forwarding of injected frames by either letting the frame analyze and forward accordingly or directly specifying the destination set. This is controlled through the BYPASS field in the CPU injection header.

Frames injected from a CPU (internal or external) can under some circumstances be sent back to the CPU. This can happen when injected frames are analyzed and hit a filter in the classifier or analyzer that copies or redirects frames to a CPU extraction queue. This behavior can be disabled through QSYS::EXT_CPU_CFG.EXT_CPU_KILL_ENA for frames injected by an external CPU or through QSYS::EXT_CPU_CFG.INT_CPU_KILL_ENA for frames injected by the internal CPU. This is similar to source port filtering done for front ports and prevents the CPU from sending frames back to itself.

4.19 VRAP Engine

The Versatile Register Access Protocol (VRAP) engine allows external equipment to access registers in the device through any of the Ethernet ports. The VRAP engine interprets incoming VRAP requests, and executes read, write, and read-modify-write commands contained in the frames. The results of the register accesses are reported back to the transmitter through automatically generated VRAP response frames.

The device supports version 1 of VRAP. All VRAP frames, both requests and responses, are standard Ethernet frames. All VRAP protocol fields are big-endian formatted.

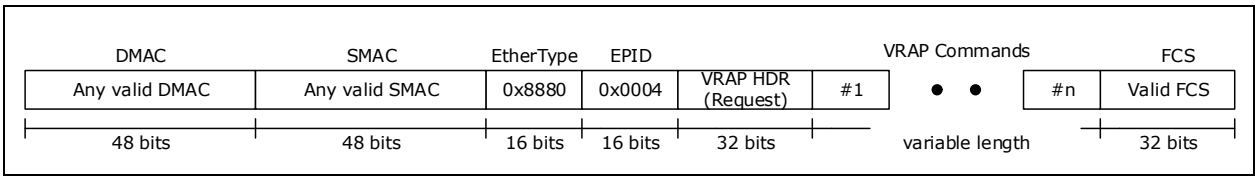
The VRAP engine processes incoming VRAP frames that are redirected to the VRAP CPU extraction queue by the basic classifier. For more information about the VRAP filter in the classifier, see Section 4.9.6, CPU Forwarding Determination.

The VRAP engine is enabled by allocating one of the two CPU ports as a dedicated VRAP CPU port (DEVCPU_QS:XTR_GRP_CFG:MODE). The VRAP CPU extraction queue (ANA:CPUQ_CFG2:CPUQ_VRAP) must be mapped as the only CPU extraction queue to the VRAP CPU port (QSYS:CPU_GROUP_MAP:CPU_GROUP_MAP). In addition, the VRAP CPU port must disable the use of CPU injection and CPU extraction headers (SYS:PORT_MODE:INCL_INJ_HDR and SYS:PORT_MODE:INCL_XTR_HDR).

4.19.1 VRAP REQUEST FRAME FORMAT

Figure 4-32 shows the format of a VRAP request frame.

FIGURE 4-32: VRAP REQUEST FRAME FORMAT



VRAP request frames can optionally be VLAN tagged with one VLAN tag.

The EtherType = 0x8880 and the Ethernet Protocol Identifier (EPID) = 0x0004 identify the VRAP frames. The subsequent VRAP header is used in both request and response frames.

The VRAP commands included in the request frame are the actual register access commands. The VRAP engine supports the following five VRAP commands:

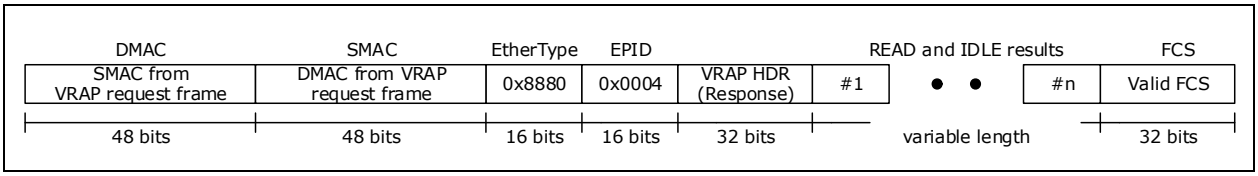
- READ: Returns the 32-bit contents of any register in the device.
- WRITE: Writes a 32-bit value to any register in the device.
- READ-MODIFY-WRITE: Does read/modify/write of any 32-bit register in the device.
- IDLE: Does not access registers; however, it is useful for padding and identification purposes.
- PAUSE: Does not access registers, but causes the VRAP engine to pause between register access.

Each of the VRAP commands are described in the following sections. Each VRAP request frame can contain multiple VRAP commands. Commands are processed sequentially starting with VRAP command 1, 2, and so on. There are no restrictions on the order or number of commands in the frame.

4.19.2 VRAP RESPONSE FRAME FORMAT

The VRAP response frame follows the VRAP request frame in terms of VLAN tagging: If the VRAP request was tagged, the VRAP response is also tagged using the same VLAN. Figure 4-33 shows the response frame.

FIGURE 4-33: VRAP RESPONSE FRAME FORMAT

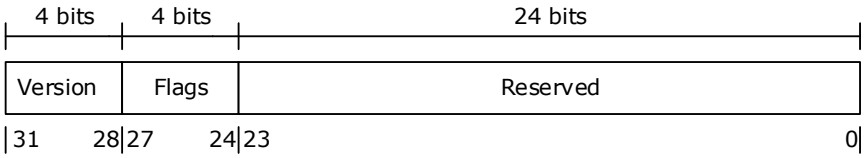


Only the READ and IDLE commands require data to be returned to the transmitter of the VRAP request frame. However, even if the VRAP request frame does not contain READ and IDLE commands, a VRAP response frame is generated with enough padding data (zeros) to fulfill the minimum transfer unit size.

4.19.3 VRAP HEADER FORMAT

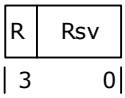
Both VRAP request and VRAP response frames contain a 32-bit VRAP header. The VRAP header is used to identify the protocol version and to differentiate between VRAP requests and VRAP response frames. The device supports VRAP version 1. A valid VRAP request frame must use Version = 1 and Flags.R = 1. Frames with an invalid VRAP header are discarded and not processed by the VRAP engine. Figure 4-34 shows the VRAP header.

FIGURE 4-34: VRAP HEADER FORMAT



Version:
Set to 0x1 in VRAP response frames.
VRAP request frames are ignored if Version <> 1

Flags:
4 bits are used for Flags. Only one flag is defined:



R: 0=Request, 1=Response
Rsv: Set to 0 in VRAP response frames and ignored in VRAP request frames.

Reserved:
Set to 0 in VRAP response frames and ignored in VRAP request frames.

4.19.4 VRAP READ COMMAND

The VRAP READ command returns the contents of any 32-bit register inside the device. The 32-bit read-data result is put into the VRAP response frame.

The READ command is 4 bytes wide and consists of one 32-bit address field, which is 32-bit aligned. The 2 least significant bits of the address set to 01. Figure 4-35 shows the request command and the associated response result.

FIGURE 4-35: READ COMMAND

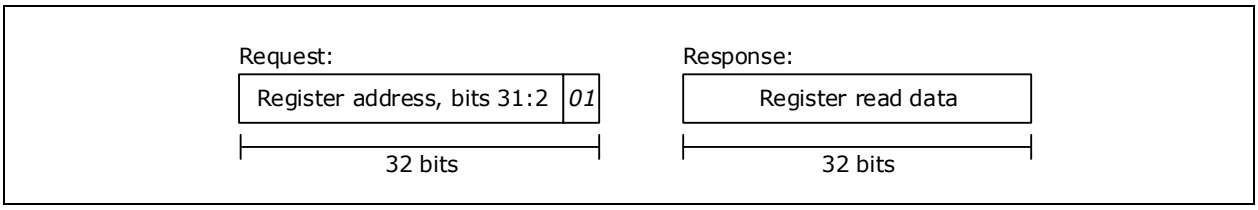
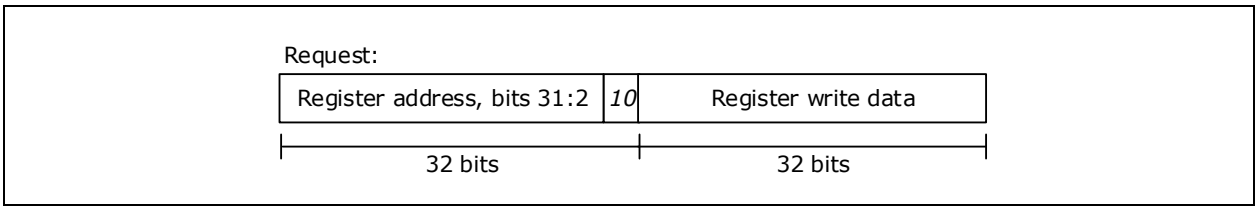


FIGURE 4-36: WRITE COMMAND

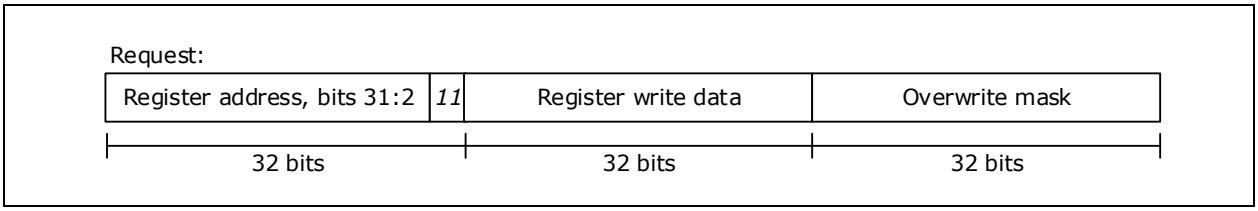


4.19.5 VRAP READ-MODIFY-WRITE COMMAND

The READ-MODIFY-WRITE command does read/modify/write-back on any 32-bit register inside the device.

The READ-MODIFY-WRITE command is 12 bytes wide and consists of one 32-bit address field, which is 32-bit aligned. The two least significant bits of the address set to 11 followed by one 32-bit write-data field followed by one 32-bit overwrite-mask field. For bits set in the overwrite mask, the corresponding bits in the write data field are written to the register while bits cleared in the overwrite mask are untouched when writing to the register. Figure 4-37 shows the command.

FIGURE 4-37: READ-MODIFY-WRITE COMMAND

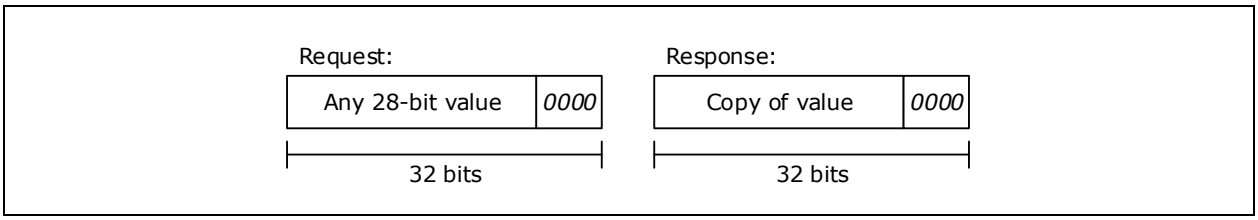


4.19.6 VRAP IDLE COMMAND

The IDLE command does not access registers in the device. Instead it just copies itself (the entire command) into the VRAP response. This can be used for padding to fulfill the minimum transmission unit size, or an external CPU can use it to insert a unique code into each VRAP request frame so that it can separate different replies from each other.

The IDLE command is 4 bytes wide and consists of one 32-bit code word with the four least significant bits of the code word set to 0000. Figure 4-38 depicts the request command and the associated response.

FIGURE 4-38: IDLE COMMAND

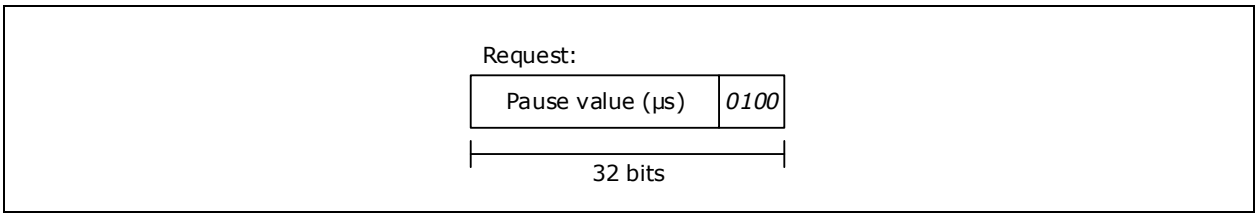


4.19.7 VRAP PAUSE COMMAND

The PAUSE command does not access registers in the device. Instead it causes the VRAP engine to enter a wait state and remain there until the pause time has expired. This can be used to ensure sufficient delay between VRAP commands when this is needed.

The PAUSE command is 4 bytes wide and consists of one 32-bit code word with the four least significant bits of the code word set to 0100. The wait time is controlled by the 28 most significant bits of the wait command. The time unit is 1 us. Figure 4-39 depicts the PAUSE command.

FIGURE 4-39: PAUSE COMMAND



4.20 Hardware Time Stamping

4.20.1 TIMING OVERVIEW

The device supports hardware-assisted PTP with high-accuracy time stamping logic and logic for adjusting the local TOD (time-of-day) counter to synchronize to an external timing server through GPIO signals. Time of day is maintained in a counter consisting of 48 bits of seconds, 30 bits of nano seconds, and 8 bits of sub-nanoseconds. This counter set is called a Local Timing Clock - or LTC in the remainder of this description.

There are 4 GPIOs which can be used by the LTCs. These are enabled by the GPIO mux configuration registers and are, in this section, described as 4 available pins. When configuring access to these pins in the PTP configuration space, they are indexed according to the PTP0-3 index numbers specified in the GPIO mux section.

The protocols used to transfer timing information across the network is defined by IEEE1588v2 (PTP) and IEEE802.1AS (gPTP). The PDUs associated with these protocols are, to a large extent, identical. References to fields in the descriptions may vary slightly between the two standards, but use of the fields are the same.

The device provides three independent LTCs that can be used for independent clock distributions. Through the timing protocols, software can determine how fast time is running in terms of what should be added to the TOD counter each core clock cycle and what the absolute value should be at any time to be synchronous to a server clock node.

The device can process protocol messages in hardware through automatic generation of delay request and sync messages.

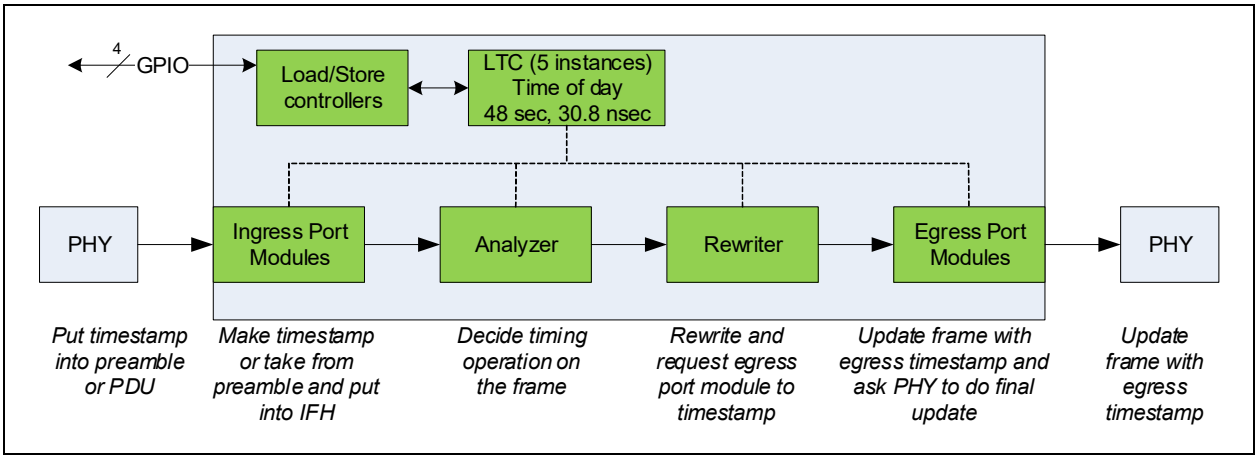
Timestamping is performed when frames are entering or leaving the switch by sampling one of the LTCs. The device supports timestamping accuracy in the sub-nanosecond range - depending on interface type.

Frame analysis is used to recognize the PTP frames that need to be processed, which is performed by either modifying the frame with timing information during passage (one-step operation) or storing timestamps for software processing (two-step operation). This frame analysis is part of the device's general frame analysis capabilities and recognizes timing PDUs under any generally supported encapsulation.

Frame modifications are done in the rewriter. When processing timing PDUs, the ingress timestamp (stored in the IFH) and the egress timestamp (when the frame will leave the device) are provided. Because the PHYs are timing aware, it is possible for them to take part in the processing.

Frame time stamping is done and stored as indicated in Figure 4-40.

FIGURE 4-40: PTP FRAME PASSAGE



There are various modes of frame timestamping, configured individually per port and direction.

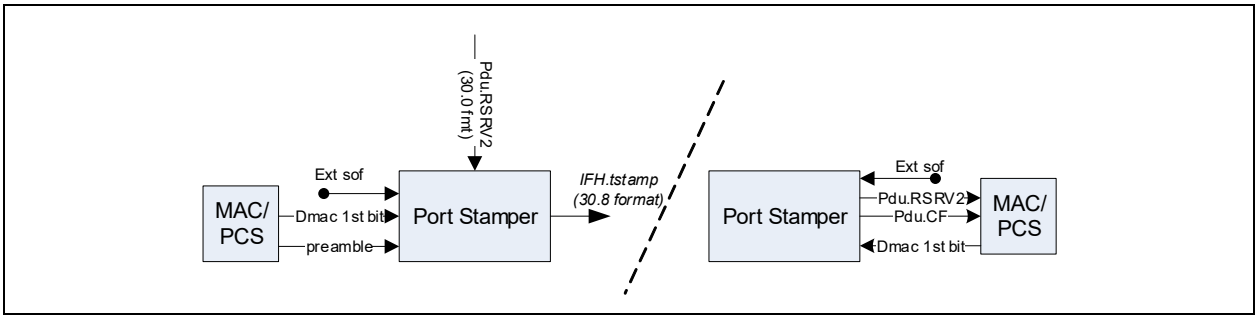
The Ingress timestamp is always stored in the IFH stamp field, and is either:

- A sampling of an LTC when the MAC indicates start of frame (EXTSOF option is for internal PHYs only)
- A value written into the RSRV2 field by the attached PHY
- A value written into the preamble by the attached PHY (PCH mode)

Egress timestamping is used by the rewriting logic, and is a sampling of the SOF indication from the MAC. The rewriter can use it for:

- Updating the correction field
- Writing it into the RSRV2 field for an external PHY to perform the final correction field update

FIGURE 4-41: PORT TIME STAMPING



4.20.2 PORT TIMESTAMPING CONFIGURATION

A port used for timestamping functions must be configured according to its speed and mode. This is performed via two registers per port: PHASE_DETECTOR_CTRL(0+1):PHAD_CTRL, and PTP_MISC_CFG. Fields of interest are seen in Table 4-42.

TABLE 4-42: PORT TIMESTAMPING REGISTERS

Mode	DIV_CFG	RX_STAMP_SEL	TX_STAMP_SEL
1000BASE-T	4	0	3
10/100BASE-T	2	1	2
1000BASE-X	3	0	3

TABLE 4-42: PORT TIMESTAMPING REGISTERS (CONTINUED)

Mode	DIV_CFG	RX_STAMP_SEL	TX_STAMP_SEL
10/100BASE-X FDX	3	0	1
10/100BASE-X HDX	3	0	3
2500BASE-X	7	0	3

In all modes, the PHAD_ENA must be set. For software that looks for a status output from timestamping engines, the LOCK_ACC can be set to 3, and is set up as a jitter tolerance in the algorithm to provide the PHAD_LOCK status.

Note: The PTP timestamping accuracy on 10Mbps RMII/RGMII ports is up to 4 us.

4.20.3 MESSAGE FORMAT

The messages processed by the device is as shown in Table 4-43. The PTP PDUs will be encapsulated under an eth-type encapsulation or within a UDP message and recognized by the VCAP rules for those frame types added to match criteria related to the PTP contents (qualified by ethtype or UDP port numbers).

The fields from the PDU that can be checked or modified by the hardware are as indicated. The description here is based on the underlying standards and a more generalized internal representation in the device.

TABLE 4-43: PTP MESSAGE FIELDS

Field	Byte Offset	Description	Match	Use	Update
MajorSdoId	0 (nib)	Protocol version info			
MessageType	0 (nib)	Message type	X		
minorVersionPTP	1 (nib)	Protocol version info			
versionPTP	1 (nib)	Protocol version info	X		
messageLength	2	Length of gPTP message			
domainNumber	4	PTP domain identifier	X		
minorsDoId	5	Protocol version info			
Flags	6	Bits 9, 10, and 15 matched (twostep flag incl)	X		
correctionField	8	Fine adjustment of master time with unit 2^-16 ns		X	X
RSRV2 (name has changed in std versions, but in this description it is called RSRV2)	16	Used for external timestamping		X	X
SourcePortIdentity	20	Clock identifier			X
SequenceNumber	30	Sequence number			X
controlField	32	not used by hardware			
logMessageInterval	33	not used by hardware			
orgtime1	34	Origin time by server			X
orgtime2	44	Origin time by server			X
cumulativeRateRatio	54	Server ppm clock offset		X	X

The orgtime1 and orgtime2 fields are generalized field names. They are only used when certain hardware actions are triggered. The VCAP rules determine whether the timing field is defined when matching against MessageType. The cumulativeRateRatio is only defined for gPTP PDUs and assumed available when the ptp domain and message type fields matches. It will only be accessed when the match rules enables operation on that field.

4.20.3.1 Software Frame Transfers

The CPU can receive copied or redirected frames as decided by the analyzer. All frames will have their timestamp (nanoseconds and sub-nanoseconds) as a field in the IFH. Through the rewriting options (the Rx command – see Section 4.20.6, "Timing Action"), the PDU can also have the correction field updated to the time it passes through the

rewriter and have the RSRV2 field set to the local time at which it was updated. This is used for reception of SYNC and DELAY_REQ messages. In general, frames directed to the CPU can have the raw frame transferred or any rewritten version of the frame, as desired by the software methodology.

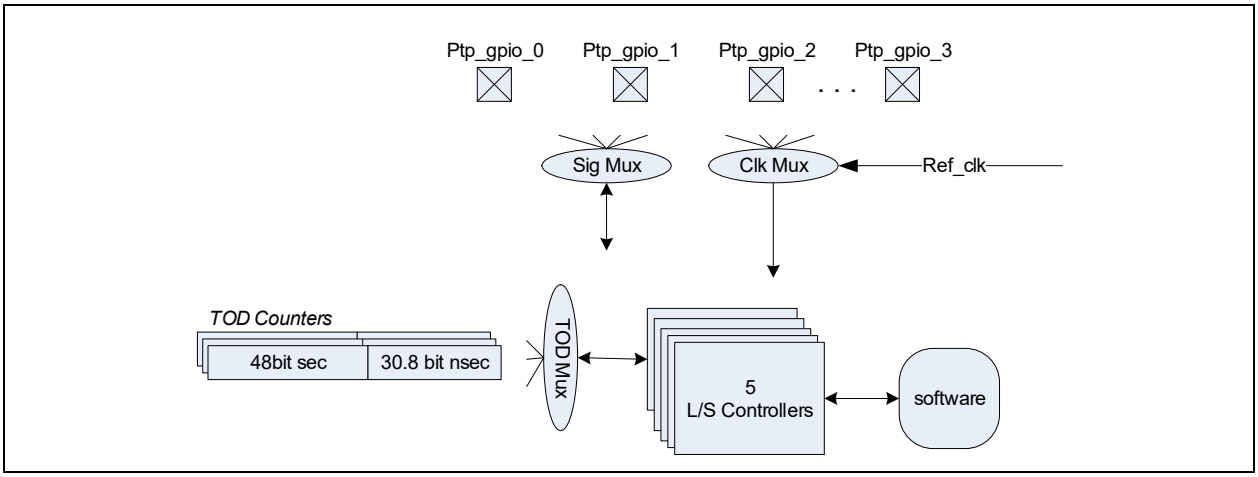
Frames transmitted from the CPU can also undergo any rewriting on the way towards the egress port. Software can fill in a 1588 PDU and transmit it with a timestamp in the IFH, indicating what time it sampled the TOD and asking the hardware to update the correction field on the egress port.

4.20.4 TIME-OF-DAY WATCHES (LTCS)

The device contains three LTCs, each telling the time in a clock distribution domain. They advance the time by a 5.59 fixed point number each core clock cycle. The constant is configured in DEVCPU_PTP:PTP_TOD_DOMAINS:CLK_PER_CFG. Software must configure this to what is known by design, or what can be deducted from interaction with the surroundings through the IEEE1588 protocol, GPIO signaling, or other means. The value can be changed atomically at any time.

Software access to the LTCs is done through load/store controllers. There are 5 of those in the device accessed through the DEVCPU_PTP:PTP_PINS registers. Each controller can be used to get or set a LTC immediately or when a GPIO edge is detected and they can be used to generate GPIO waveforms controlled by an LTC running value. If a GPIO is used by the controller, another GPIO can be used to synchronize the detection or generation of edges to a different clock domain.

FIGURE 4-42: TIME-OF-DAY LOAD/STORE



The operations that the controller can perform are described in Table 4-44.

TABLE 4-44: LOAD/STORE (L/S) OPERATIONS

L/S Operation Mode	Description
LOAD	An LTC is set to a configured value or added to a configured value. Operation can be performed immediately or when a selected GPIO indicates an event.
STORE	An LTC is latched into registers and is readable from software. Operation can be performed immediately or when a selected GPIO indicates an event.
CLOCK	Generate a waveform on a selected pin based on the value of a selected LTC. • Square form with specified high and low period (minimum 120Hz frequency) • Pulse with specified pulse time and width in nanoseconds • Connected to selected bit of the counter
TOD	Generate a serial stream indicating the seconds value of a LTC, at programmable value of the nanosecond part.

4.20.4.1 GPIO Inputs

When a pin is used to trigger a LOAD or STORE operation, it is done at a time close to when the event is seen. The detection of when the event happens has half a system clock cycle uncertainty when the GPIO is used asynchronously, which for a 160 MHz device is ~3ns. Storing an LTC value, when a GPIO tells the device to do so, adds 3ns noise to how close the local and external LTC are.

A GPIO can be configured to be associated with a clock domain as well, by which the detection on the event time (rising edge of a provided clock) can be done with sub-nanoseconds accuracy instead. In this mode, the load/store controller uses two GPIO inputs if the clock comes from an external device.

If it is known that the external device is synchronous to the local reference clock for the system PLL, that can also be used as a clock source.

A special embedded clock option also exists, where the clock and events are received on the same GPIO. In this case, the GPIO must show a clock with duty cycle of 40-50% and present a synchronous event by a single cycle with a duty cycle of 10-40%. This is used when embedding a PPS pulse into the PTP clock source signal in the mode named “ePPS”. A constant delay will be seen from the odd duty cycle period to the time of capture which must be accounted for in the software processing of the result.

4.20.4.2 GPIO Outputs

When using a load/store controller for generating pulse/clock output, it will be performed from the internal system clock domain. The generated signal can therefore only change every half a system cycle. This adds up to half a cycle of uncertainty to when the signal changes will be generated. Each time a rising edge of the generated signal is done, a software readable register PTP_PIN_OUTP_OFS can be read telling what the actual local time of day value was when the event was generated.

4.20.4.3 Physical I/O Delay Evaluation

A method exists which allows the PIN_IOBOUNCH_DELAY register to indicate the echo time from internal logic to and from the selected GPIO pad cell, including the output driver and input buffer. This can be used for fine adjustment when synchronizing two systems. The GPIO signal delay may vary over temperature and chip process.

Depending on the capacitive loading of the GPIO output, the echo time will typically be around two times higher in the output direction than the input direction, so that the internal and GPIO input delay is 33% of the measured echo time and the internal and GPIO output delay each are 66% of the measured echo time.

To evaluate the delay:

1. Set the pin of interest to form a synced 1pps with WFL=all ones.
   a) Set PIN_WF_LOW=all ones
   b) Set PTP_PIN_SYNC=1
   c) Set PTP_PIN_ACTION=3
2. Set the pin operation to UNDEFINED
   a) Set PTP_PIN_ACTION=7
3. Embed the external reference clock onto the pin
   a) Set PTP_PIN_EMBEDDED_CLK=1
   b) Set PTP_CLK_SELECT=5
4. Read PIN_IOBOUNCH_DELAY (IOB_VAL)

The echo delay can by this be calculated as:

EchoDelay = (period of the reference clock) x IOB_VAL x 2-17.

For example, with a 25MHz reference clock and a readout of 6918:

EchoDelay = (40 ns) x 6918 x 2-17 = 2.11 ns

4.20.4.4 Multi-Chip Systems

Having multiple chips involved in the system solution adds the need to synchronize each chip’s counters. This is done by the methods described in the previous sub-sections. Depending on available clocks, shared reference clocks between chips can be constructed. This is outside the scope of this document.

In the remainder of Section 4.20, Hardware Time Stamping, it is assumed that switch and PHYs are synchronized by some means, and as such can be viewed as referring to the same counter no matter what chip it is used in.

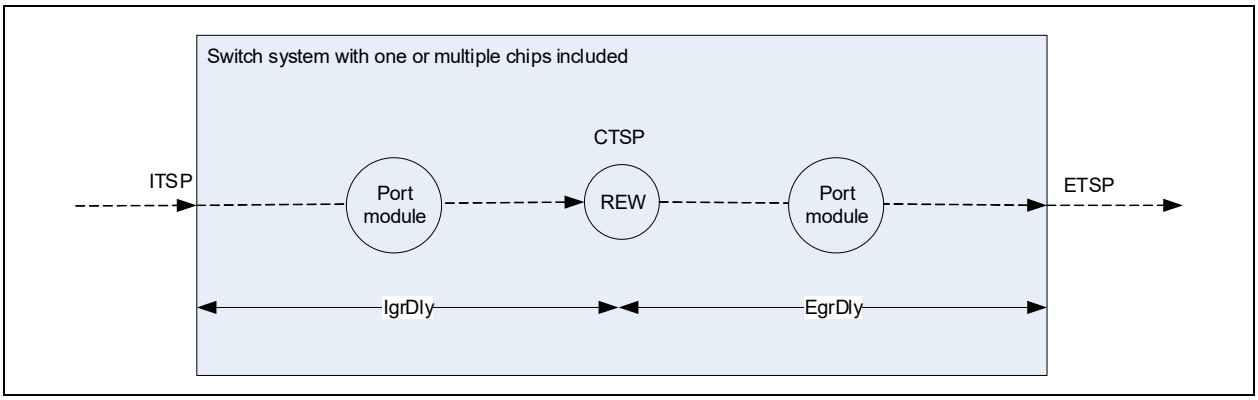
4.20.5 FRAME ANALYSIS

The frame analysis module can recognize PTP frames under any encapsulation generally supported by the VCAP look-ups. The available commands are described in Table 4-45.

TABLE 4-45: TIMESTAMP POINTS

Term	Description
Ingress TS Point, ITSP	Point in the system where the ingress time stamp is to be generated. It can be the time stamping block in a 1588 aware PHY or the SerDes interface on the switch device.
Egress TS Point, ETSP	Corresponding point in the egress direction.
Central TS Point, CTSP	The point inside the device where the rewriter processes the first bit of the DMAC.
IgrDly	Ingress delay = the time passed between ITSP and CTSP. Time is measured by the LTC the ingress port has been assigned to in PTP_CFG.PTP_DOM in the port module. It will by default be LTC 0 for all ports.
EgrDly	Egress delay = the time passed between CTSP and ETSP. Time is measured by the LTC for the domain the egress port has been assigned to in PTP_CFG.PTP_DOM in the port module. It will by default be LTC 0 for all ports.
TSC	Value of the LTC the frame is assigned to use by the action vector (described below).

FIGURE 4-43: TIMESTAMPING POINTS AND TERMS



For the time stamping points on the I/O interfaces, there is a constant path delay between the physical SerDes interface and the timestamping logic in the port module. This delay must be configured by software at link up, as it usually has variations due to internal states at link up. The delay consists of pipeline delays from the timestamping logic to the I/O, as well as dynamic delays from the deserialization process (barrel shifting).

Receiving frames on a serial link is performed through a de-serialization process. A number of bits are handled in parallel, and the clock domain for the parallel data is divided down from the bit clock. That clock divider will come up in a new state at link up and is not correlated to the incoming data stream. The phase between the 1st bit of a frame and the clock domain can be found by reading out barrel shifter states, which tells how many bit times there are between the first bit of a frame and the receive domain clock edges.

The pipeline delay contribution depends on port mode and speed, thus the mode-to-delay table must be available in software. The exact values are described in tables provided outside this document.

4.20.6 TIMING ACTION

The timing action is described in a 16-bit vector coming from the VCAP IS2 lookup result REW_CMD. This format is described in Table 4-46. Each 1588 frame passing through the analyzer will be matched against the VCAP rules and processed according to the role of the device and ports involved.

TABLE 4-46: PTP UPDATE ACTIONS

Bits	Name	Description	
15:10	gPTP Options	See Table 4-51, "gPTP VCAP Action".	
9:8	Advanced Options	Enables advanced options, which in this device is only defined for value 01, enabling gPTP processing.	
7:6	Time Domain	Selects the LTC and TSC should be based on.	
5	SMAC Update	Change SMAC to PTP_SMAC_LOW/_HIGH before transmission.	
4:3	Add	Add some content to the CF field, independent of comman	
		00	-
		01	Add REW::PTP_EDLY_CFG(dst)
		10	Add REW::PTP_IDLY1_CFG(src)
		11	Add REW::PTP_IDLY2_CFG(src)
2	SeqNumUpd	Updates sequence number in PDU. 512 sequence number counters are available in hardware. Which one to use and update must be set by software in the PDU.seq_-num(8:0) field before transmission. Setting PDU.seq_num(9)=1 will skip the counter table update.	
1:0	Command	00 [NOP]	Don't perform any timing update.
		01 [RESI]	CF += IgrDly + EgrDly
		10 [RX]	CF +=IgrDly. Rsrv2=TSC
		11 [TX]	CF = EgrDly, ORG=TSC

The other features are triggered by REW_CMD action:

TABLE 4-47: REW_CMD FEATURES

Function	RewCmd Encoding	Description
TWOSTEP FIFO	0x0004	Ingress and egress timestamp will be stored in the timestamp FIFO for readout from software.
HAIRPIN	0x0040	SMAC and DMAC will be swapped before transmission.

The PTP protocols are not described in this document. The basic roles and actions are briefly summarized in the following sub-sections:

- Section 4.20.6.1, Transparent Clocks
- Section 4.20.6.2, Clock Client

4.20.6.1 Transparent Clocks

For one-step transparent clocks, the correction field of Sync and Delay_Req messages must be added to the residence time in the device. This is done by the RESI command, potentially with an Add request for adding path or asymmetry delays as well depending on E2E or P2P delay measurement. If needed, the SMAC update flag can be used to replace the source MAC address of the frame with a per port configured value.

A server must send out SYNC frames to its clients with its LTC value upon departure in the ORG+CF fields. That can be done by having the AFI to transmit the same frame periodically asking for the TX command, and through the SeqNumUpd get the sequence number updated. Requests for this must be set in the IFH upon injection. Please read the IFH description for details.

The frame from the source (the CPU here) must have the sequence number in the injected PDU prefilled with an index, which then points into the REW:PTP_SEQ_NO table. That table entry contents will be filled into the PDU before transmission, and the table entry will be incremented. If pointing beyond the number of entries (512), the selection will be modulated by 512, and the table entry will keep its current value.

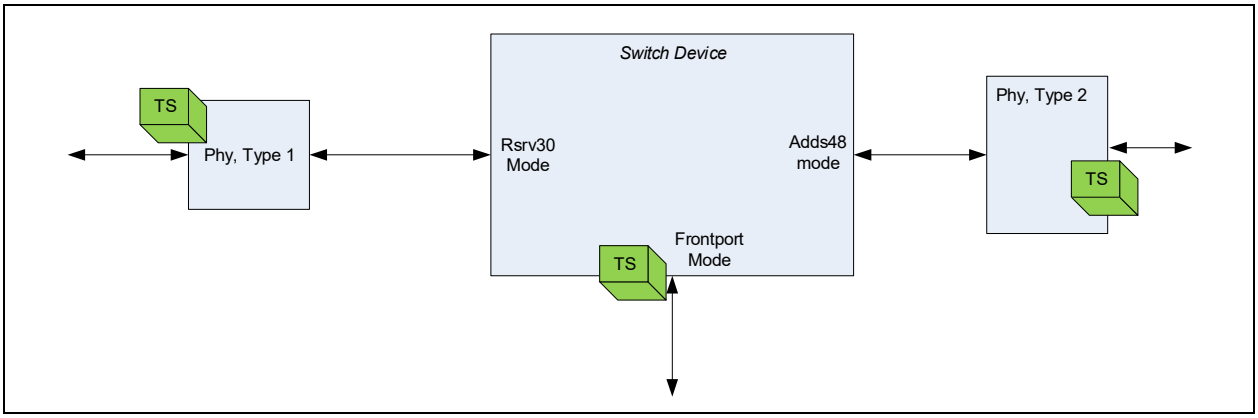
4.20.6.2 Clock Client

A clock client must receive SYNC and delay response messages, and store what the local time was upon reception. That is performed via the RX command.

4.20.7 PORT TIMESTAMPING MODES

The rewriter has per switching port a configuration which tells how the ingress and egress delays are found for frames coming from or going towards the port. The mode basically defines where the ITSP and ETSP are located when frames are coming or going to the port. The mode can be FRONT, RSRV30, or ADDS48.

FIGURE 4-44: PTP PORT MODES



The modes define how the operations “+IgrDly” and “+EgrDly” are carried out. It uses the following pieces of timing info:

- IFH.TS: The ingress timestamp
- LTCI: The LTC value for the domain the ingress port is assigned to
- LTCE: The LTC value for the domain the egress port is assigned to
- TodE: The value of the LTC for the egress port when 1st bit is transmitted

TABLE 4-48: INGRESS DELAY CALCULATION

Mode	PHY Does	Port Module Does	Rewriter Does
FrontPort	-	IFH.TS = LTCI	CF + LTCI - IFH.TS
Rsrv30	RSRV2 = LTCI	IFH.TS = RSRV2	CF + LTCI - IFH.TS
Adds48	CF -= LTCI	-	CF + LTCI

TABLE 4-49: EGRESS DELAY CALCULATION

Mode	Rewriter Does	Port Module Does	PHY Does
FrontPort	CF += $LTCE_{eport}$ - LTCE	-	-
Rsrv30	CF += $LTCE_{eport}$ - LTCE RSRV2 = $LTCE_{eport}$	-	CF += LTCE - RSRV2
Adds48	CF -= LTCE	-	CF += LTCE

The Tod values in the tables are the time-of-day values from a LTC, which includes 48-bit seconds, 30-bit nanoseconds and 8-bit sub-nanoseconds. The rewriter ingress and egress operation operates simultaneously.

The following are examples for the cases where port modes and LTC domains are equal, for the +IgrDly+EgrDly:

Front: CF+= ($LTC_{rew} – LTC_{iport}$) + ($LTC_{eport} – LTC_{rew}$) = $LTC_{eport} – LTC_{iport}$

Rsrv30: CF+= ($LTC_{rew} – LTC_{iphy}$) + ($LTC_{eport} – LTC_{rew}$) + ($LTC_{ephy} – LTC_{eport}$) = $LTC_{ephy} – LTC_{iphy}$

Adds48: CF += ($LTC_{rew} – LTC_{iphy}$) + ($LTC_{ephy} – LTC_{rew}$) = $LTC_{ephy} – LTC_{iphy}$

For the RSRV30 port mode, only the nanoseconds portion is used. For the ADDS48 modes, the LTC value is translated into the lower 64 bits of the number of 2^{-16} nanoseconds since time 0.

Val64 = ((10^9 x seconds + nanoseconds) mod 2^{48}) x 2^{16} + subnanosecs x 2^8

4.20.8 MULTIPLE DOMAINS

The device can be operated with one or more time of day domains, used for the 1588 protocol, and other time domains for the timing interaction between PHYs and the switch device. For all frame operations, up to three domains are relevant, even though they may all use the same TOD counter

- **Ingress domain:** Time domain used to calculate the IgrDelay. The domain number must be configured in the port module in use and the central rewriter. Those two configurations must be identical for the port and rewriter to understand the time information passing between them.
- **Egress domain:** Similar to the ingress domain, this is used for EgrDelay calculations
- **PDU domain:** This is the time domain the PDU itself should be based on when filling out the ORG/RSRV2 fields in Rx and Tx commands.

Multiple domains can be used for having a single chip be the timing master for multiple independent vLans, and it can be used for distributed transparent clocks where the residence time between neighbors to the local device is calculated with a reference timer other than the master TOD it synchronizes.

The applications of having multiple running time domains are outside the scope of this document.

4.20.9 TWO STEP OPERATION

If the command in the action vector is TWOSTEP, ingress timestamp from the IFH and the egress time stamp from transmission are stored in a timestamp FIFO, which can be read out using a software action. Each frame passing through with the action will get its ingress and egress timestamps stored together with the physical ports the frame passed through.

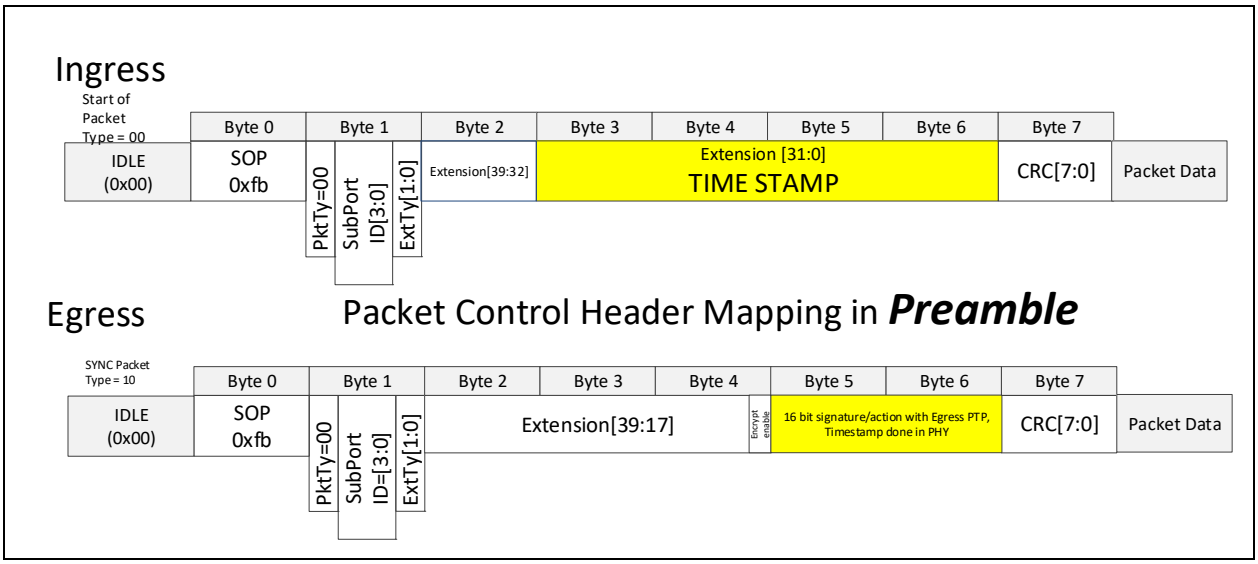
For server operation, the CPU can inject a frame with IFH.TS set to any value and ask for TWOSTEP storing of the time stamps. By reading out the FIFO, the egress time can be found by matching against the artificial IFH.TS the frame was marked with.

The time stamps stored in the two-step FIFO are the (nanosec, subnanosec) values at the time of 1st bit transmitted. To find the full TOD at the event, software may read out the current TOD and append the timestamp with the seconds value at the time of event.

4.20.10 PCH/MCH ENABLED PHYS

If a PHY supports timing information transferred through PCH headers, the preamble is replaced with timing information for the frame. The format is defined in the USXGMII specification and extended by Microchip to allow for requesting one-step updates performed by the PHY.

FIGURE 4-45: PCH/MCH FORMATS



4.20.10.1 Ingress Processing

If enabled in the port module, timestamping information from the PCH can replace the port ingress timestamp.

The ingress processing of the format is simple:

- The TS information is extended with the local TOD counter. If the PCH format is set to 16.16 format, the 14 most significant bits from the TOD counter is appended the TS, to feed a 30.8 timestamp into the core as for front ports.
- The sub port ID is checked against a device configured value. If there is no match, a sticky error flag is set but data processing is unaffected.
- The PCH checksum (bits 7..0) is checked, and a sticky error is set if it fails.

Ports using PCH for ingress time stamping must in the rewriter be ingress configured as a FRONT port.

4.20.10.2 Egress Processing

The egress processing generates a PCH/MCH preamble instructing for either having the PHY to do nothing, to do the final CF update of a frame, or to store an egress timestamp locally in the PHY. This is encoded in the 40-bit extension field:

Do Nothing: 0x00.00.00.00.00

Twostep: 0x00.80.00.00.00 + signature

Onestep: 0x00.00.00.80.18

+ (checksum_upd?0x200:0)

+ (offset in 16b unit from 1st bit in preamble)

When the PHY receives a onestep update request, it must add its LTC converted to a 48.16 fixed point nanosecond value onto CF.

Ports using PCH for egress must have their egress mode configured to ADDS48.

4.20.10.3 MACSec Support

As a side function to the above functionality, the PCH header can get its EncryptEna bit set (LSB of byte 4) if enabled by the PTP_PCH_TX_ENA configuration in the port module. A mode here also exists where specific software generated frames can bypass the encryption request. The encryption is carried out by an attached PHY as the switch role is only to ask for that translation through the PCH flag.

4.20.11 GPTP SUPPORT

The IEEE802.1AS standard adds on top of IEEE1588 handling, a mode where the local LTC is not adjusted to run with a speed indicated by the server. Instead, a node must be aware of its local ppm difference relative to the server. This short description briefly describes the main principles of how a server can synchronize clients in a network of gPTP nodes.

Through pdelay messages (handled in software), the ppm difference to link partners are evaluated and stored in hardware, as neighbor rate ratio values (I.e., clock runs 27 ppm faster than the PTP device connected to port 5). This is referred to as NRR.

When SYNC or FOLLOWUP messages are received from a server, the PDUs contain a field with the PPM difference to the server originating the packet, the cumulativeScaledRateOffset (RR in the following). A hardware table will be updated with the effective RR of the server compared to the local clock, by adding the ingress port specific NRR to the incoming RR, which then becomes our nodes relative speed compared to the server. This is stored in a 4-entry server table for use during frame transmissions and residence time calculations.

When sending SYNC frames from this device, it will look up the RR from the server table and update the cumulativeScaledRateOffset field in the frame. The correctionfield with added residence time, as measured by the local LTC, will use the detected RR to adjust the measured residence time with the ppm difference we have to the server, which is the source of how time evolves.

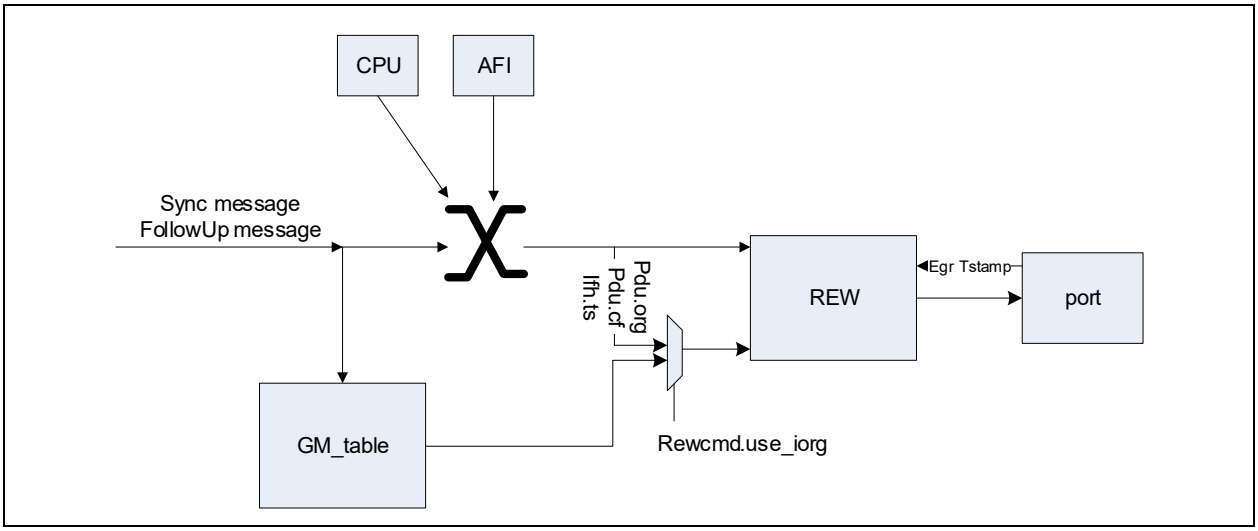
It is possible to keep the server table untouched through the vcap action flags and keep it controlled by software, as all fields are mapped into CSR accessible registers.

4.20.12 GPTP FORWARDING

When frames from a gPTP server are received, they can be directly forwarded to slave ports as it is done for 1588 operation. However, in the gPTP model, all ports are boundary clocks with independent incoming and outgoing frame rates as well as expected resilience to frame loss. The recommended mode of gPTP operation is therefore to terminate incoming SYNC/FOLLOWUP frames by not forwarding them anywhere, and have the AFI to transmit frames using information from the server table (see Figure 4-46).

In the following, FOLLOWUP messages are not specifically mentioned, as they can be treated as SYNC frames. They differ in the message type field in the PDU, and can by that be matched with each a VCAP rule.

FIGURE 4-46: GPTP OPERATION



There are three basic modes of operation supported, as depicted in Figure 4-46:

- Forward frames to slave ports when they are recieved and update residence time based on frame contents and ingress timestamp. This assures short residence time and minor updates to the CF field. It does not guarantee resilience to frame loss. The mode is accomplished through the rew_cmd.use_iorg=0, and by not killing the frame from forwarding in the analyzer. The RESI should be used here, with selected delay settings in the DlySel field of the rew_cmd.
- Terminate incoming frames and store cf, org, and ingress timestamp in the GM_table table. Configure the AFI to send out the same frame periodically, asking for update of the sequence number and use the gm_table contents for timing updates. This mode does not guarantee short transition times for SYNC/FOLLOWUP messages, if that is desired despite the general boundary clock model of operation assumed by gPTP networks. The rew_cmd for capturing the incoming frames have the use_iorg and use_irr set, and the command set to NOP. The AFI generated frames should use the RESI command, use_irr=1, use_iorg=1.
- Forward all frames directly to slave ports, update and use the GM_table, and forward the frame as well to the CPU. Let software have some kind of watchdog for telling that a SYNC frame apparently has gone missing, and let the CPU inject a frame with the missing sequence number, using the GM_table for filling out timing information. The rew_cmd should here have use_irr=use_iorg=1, command=RESI.

4.20.13 GPTP SERVER TABLE

When a VCAP REW_CMD action enables gPTP mode, it will point to one of four main records in a hardware table, which contains the fields outlined in Table 4-50.

TABLE 4-50: GPTP SERVER TABLE

Field	Description
iTS	Ingress timestamp of the latest SYNC frame received from the master
iCF	Correction field value of the latest SYNC frame
iORG	ORG field of latest SYNC frame received
RateRatio	Latest Rate Ratio value for the gPTP domain
Clock ID (CFG)	Clock identifier to use when generating frame on this gPTP domain
Tod Domain (CFG)	LTC index to use within this gPTP domain

The available REW_CMD action vector additions are detailed in Table 4-51.

TABLE 4-51: GPTP VCAP ACTION

BitPos	Name	Description
14:13	tbl_idx	Selects GM table
12	Use_spid	Replace clock identity in transmitted frames
11	use_rr	Update and use RateRatio
10	use_org	Update and use ORG field

The master table contains the latest information received from a master, and a background process updates that information every 512 ns to make it independent on when it is used for generating frames. Detected master rate ratio is used in this process.

4.20.14 RATERATIO MATH

The cumulativeScaledRateOffset is a field in a TLV following the gPTP header. It has a fixed offset from the start of the PDU (offset 54), and is a signed 32bit number RR representing a ppm difference of the transmitting partners clock relative to the originating grand master node.

$$gm\_ofs = RR*2^{-41}$$

That provides a ppm span of +/- 1000 ppm, and a unit of 2^{-41}. The Neighbor rate ratios are measured and stored with same definition and resolution.

$$nb\_ofs = NRR*2^{-41}.$$

When an incoming frame tells that it comes from a node with some RR, it means that the transmitting node has a clock period which is:

$$CLKPER_{in} = CLKPER_{gm}*(1+gm\_ofs).$$

If the local node has detected that it runs nb_ofs faster than the node from which it got the frame, it runs with a period

$$CLKPER_{local}=CLKPER_{in}*(1+nb\_ofs) = CLKPER_{gm}*(1+gm\_ofs)*(1+nb\_ofs)$$

The local device RR to the grand master therefore is:

$$ofs\_{local} = (1+gm\_ofs)*(1+nb\_ofs)-1 = 1+gm\_ofs+nb\_ofs+nb\_ofs*gm\_ofs$$

The last contribution is less than 1ppm due to the range limits (1000ppm^2), and is disregarded.

If we assume that ppm offsets at most can become 100 ppm, the contribution become at most 0.01ppm.

When a residence time is measured in the local device, it is done by timestamping in the local time domain, where a residence time of T is measured, and must be added ofs_{local}*T. For frames directly transferred without using the AFI, that T can be assumed to be less than 10us. Adding 0.01ppm to that is 100fs, which then is irrelevant to the overall system accuracy.

If however the AFI is used to generate frames, and the system receives 8 frames /sec from a server, the delay from latest received server SYNC frame to the generation of a frame going out can be up to 125ms, and even more if some frames are lost. Adding 0.01ppm to that approaches the range of 1 ns. If this is unacceptable, a higher frame rate must be used.

The master table is automatically advancing the received timing information every 512 ns as measured by the local LTC. That is done by adding 512 ns to the iTS in the table, and adding 512ns*(1+ofs_local) to the iCF.

$512 * (1+ofs_{\_local})= 512 * (1+RR_{local} * 2\text{-}41)= 512 + RR_{\_local}*2^{-32}$.

As the CF field has a resolution of 2^{-32} ns in the table, we introduce no error by this scheme.

4.20.15 MISCELLANEOUS TIMING HOOKS AND HANDLES

This section provides a quick overview of miscellaneous features related to timing functionality.

TABLE 4-52: GPTP SERVER TABLE

Feature	Description
High Accuracy Configurations	By default, timestamping is performed with the accuracy defined by a 330 MHz sampler, thus providing an accuracy of around 3 ns. Each timestamping point in the device has a phase detector added, which can improve this accuracy down to 50-150 ps, provided the clock at which the event is seen isn't too synchronous to the sampler clock. Accuracy will depend on the level of independency the clocks have. The phase detector is available for ingress and egress timestamping on all ports, and on each load/store controller for detecting incoming PPS pulses.
Status Available	Each port module has a PTP_EVENTS register which indicates if a one-step correction field update has resulted in overflow and/or whether received PCH headers have errors. In the REW, status indicates when front port frames are recieved with the RSRV field not cleared.
Interrupts	The PTP_PIN_INTR register can be used to configure whether a CPU interrupt is generated when the Load/Store controller has executed the operation. Software can use this to react to an incoming PPS pulse.
Two-Step FIFO	In the PTP register block, the PTP two-step FIFO can be read out. It has room for 512 timestamp sets (ingress, egress), and it can be configured in PTP_OVWR_ENA whether unread stamps will be overwritten by new events, or new events will be discarded.
Time Domain Access	In the PTP_DOM_CFG register, each timing domain can be cleared, kept at current value, and read out by software.
UDP Handling	Updated UDP frames will be checksum cleared for IPv4 and get a 2-byte checksum padding field updated for IPv6. This can be disabled in the rewriter block per port.

TABLE 4-52: GPTP SERVER TABLE (CONTINUED)

Feature	Description
Sub-Nanosecond Accuracy with RSRV30 Interaction to External PHYs	In case the RSRV30 mode is used to get an external PHY to add the delta between the RSRV field value and its local timestamp, the rewriter will step back in time for the ingress delay evaluation to hit a time at an integer number of nanoseconds. If the PHY has sub-nanosecond time stamping capabilities, system accuracy can benefit even though a field in the PDU is used in the transfer without that level of accuracy.

4.21 Layer 1 Timing

There are two recovered clock outputs that provide timing sources for external timing circuitry in redundant timing implementations.

The recovered clock outputs are mapped on GPIO overlaid functions. For more information, refer to Section 3.3, GPIO Alternate Functions.

It is possible to recover receive timing from any of the Ethernet ports. For aggregated port interfaces like QSGMII links, timing must be recovered in the PHY, because the QSGMII link retimes during aggregation of data streams.

The HSIO:SYNC_ETH_CFG register provides per recovered clock divider configuration, allowing division before sending out clock frequency.

The SerDes recovered clock outputs also have individual divider configuration (through HSIO:SD[0-1]:SYNC_ETH_SD_CFG) to allow division of receive frequency. Note that division with 66/32 will result in a gapped clock that normally is not recommended for SyncE operation.

The recovered clocks are single-ended outputs. The suggested divider settings in the following tables are selected to make sure to not output too high frequency via the input and outputs. A maximum frequency of 62.5 MHz is allowed.

TABLE 4-53: SERDES RECOVERED CLOCK SETTINGS FOR 1 GE

Output Frequency	Register Settings for Output *n*, Lane *m*
25 MHz	SYNC_ETH_CFG[n].RECO_CLK_ENA=1, SYNC_ETH_CFG[n].SEL_RECO_CLK_DIV=4, and SYNC_ETH_CFG[n].SEL_RECO_CLK_SRC=(m) SD[m]::SYNC_ETH_SD_CFG.SD_RECO_CLK_DIV=0
31.25 MHz	SYNC_ETH_CFG[n].RECO_CLK_ENA=1, SYNC_ETH_CFG[n].SEL_RECO_CLK_DIV=1, and SYNC_ETH_CFG[n].SEL_RECO_CLK_SRC=(m) SD[m]:SYNC_ETH_SD_CFG.SD_RECO_CLK_DIV=0
Note: There are internal dividers for each interface and data rate.	

TABLE 4-54: SERDES RECOVERED CLOCK SETTINGS FOR 2.5 GE

Output Frequency	Register Settings for Output *n*, Lane *m*
31.25 MHz	SYNC_ETH_CFG[n].RECO_CLK_ENA=1, SYNC_ETH_CFG[n].SEL_RECO_CLK_DIV=4, and SYNC_ETH_CFG[n].SEL_RECO_CLK_SRC=(m) SD[m]:SYNC_ETH_SD_CFG.SD_RECO_CLK_DIV=1
62.50 MHz	SYNC_ETH_CFG[n].RECO_CLK_ENA=1, SYNC_ETH_CFG[n].SEL_RECO_CLK_DIV=4, and SYNC_ETH_CFG[n].SEL_RECO_CLK_SRC=(m) SD[m]:SYNC_ETH_SD_CFG.SD_RECO_CLK_DIV=0

TABLE 4-55: CUPHY RECOVERED CLOCK SETTINGS FOR 1000BASE-T

Output Frequency	Register Settings for Output *n*, Port *m*
31.25 MHz	SYNC_ETH_CFG[n].RECO_CLK_ENA=1, SYNC_ETH_CFG[n].SEL_RECO_CLK_DIV=1, and SYNC_ETH_CFG[n].SEL_RECO_CLK_SRC=(m+4)
62.50 MHz	SYNC_ETH_CFG[n].RECO_CLK_ENA=1, SYNC_ETH_CFG[n].SEL_RECO_CLK_DIV=0, and SYNC_ETH_CFG[n].SEL_RECO_CLK_SRC=(m+4)

TABLE 4-56: CUPHY RECOVERED CLOCK SETTINGS FOR 100BASE-TX

Output Frequency	Register Settings for Output *n*, Port *m*
25 MHz	SYNC_ETH_CFG[n].RECO_CLK_ENA=1, SYNC_ETH_CFG[n].SEL_RECO_CLK_DIV=6, and SYNC_ETH_CFG[n].SEL_RECO_CLK_SRC=(m+4)
5 MHz	SYNC_ETH_CFG[n].RECO_CLK_ENA=1, SYNC_ETH_CFG[n].SEL_RECO_CLK_DIV=4, and SYNC_ETH_CFG[n].SEL_RECO_CLK_SRC=(m+4)

It is possible to automatically squelch the clock output when the devices detect a loss of signal or PCS loss of lock on an incoming data stream. This can be used for failover in external timing recovery solutions. Auto squelching is enabled by setting SD[n]:SYNC_ETH_SD_CFG.SD_AUTO_SQUELCH_ENA for the SerDes ports and CHIP TOP:CuPHY CFG(n):CuPHY PORT CFG.AUTO SQUELCH ENA for the CuPHY ports. To enable auto squelch on the CuPHY port, it is also necessary to enable the Fast Link Fail mode. This is done by setting the “FLF Enable” and “Enable FLF Link Down” bits in PHY register EP 0.15 (bit 1 and 0).

The SerDes signal detect pin can also be used to squelch the recovered clock. This is enabled by setting SD[n]:SIG-DET_CFG.SD_ENA and should be used when the ports are connected to optical modules.

When squelching the clock, it will stop when detecting loss of signal (or PCS lock). The clock will stop on either on high or low level.

4.22 Clocking and Reset

The chip is reset when the **nRESET** pin is low. The chip will start booting and be operational approximately 14ms after **nRESET** has toggled high. The strapping pins are latched when **nRESET** toggles high.

The following reference clocks can be provided to the device:

- **25 MHz reference on the XI pin or XI and XO pins**
  This may be provided by either:
  - A 25 MHz clock crystal connected between the **XI** and **XO** pins
  - A 25 MHz singled-ended oscillator connected to the **XI** pin

  Note: This clock is mandatory and must be provided for the device to operate.
- **125 MHz clock on S0_CLKP pin**
  The 125 MHz clock mode is enabled by pin strapping. When SerDes is used in QSGMII mode, SerDes can be configured to use the 125 MHz clock. If 25 MHz mode is selected for both CuPHY and SerDes, the 125 MHz clock is not used and must be unconnected.

The switch core and CPU clock generator PLLs use an internal 25 MHz clock as reference. When the external reference for CuPHY is strapped for 125 MHz operation mode, the internal 25 MHz PLL reference is generated by dividing the 125 MHz clock by 5. The internal 25 MHz clock can also be supplied using the crystal (XTAL) connected to the **XI** and

XO pins or by using an oscillator connected to the XI pin. The CuPHY and SerDes can use either 25 MHz or 125 MHz reference clocks, but if any of the SerDes are running in QSGMII mode it must be configured to use the 125 MHz reference clock. Refer to Section 3.4.1.4, SerDes 125 MHz Selector (SD_125M) for details on how this is selected.

4.23 VCore System and CPU Interfaces

This section provides information about the functional aspects of blocks and interfaces related to the VCore on-chip controller system and external CPU systems.

The VCore system can control the device independently or it can support an external CPU, relieving the external CPU from bringing up and maintaining the links and the switch core.

An external CPU can be connected to the LAN9645xF device through the serial interface (SI), the I^2C client interface, the dedicated MIIM Client interface, the UART command line monitor, or through Versatile Register Access Protocol (VRAP) formatted Ethernet frames via the NPI Ethernet port. Through these interfaces, the external CPU has access to the complete set of registers in the LAN9645xF device.

Features:

- Peripherals
  - Interrupt controller
  - 2x I^2C controllers
  - 2x MIIM controllers
  - Serial IO controller
- External CPU support
  - Serial Interface in client mode
  - MIIM Interface in client mode
  - I^2C interface in client mode
  - UART command line interface
- Up to 51 GPIOs
  - Number of available GPIOs depends on package selection. See Section 3.3, GPIO Alternate Functions for further details.
  - Fully programmable through Set/Clear registers
  - Multiplexing of peripheral functions per I/O line
  - Each I/O line can be assigned to a peripheral or used as a general purpose I/O

4.23.1 EXTERNAL CPU SUPPORT

An external CPU attaches to the LAN9645xF through SI, I^2C, MIIM, UART or VRAP. Through these interfaces, an external CPU can access (and control) the device. For more information about interfaces and pin mapping see Section 3.3, GPIO Alternate Functions.

4.23.1.1 Serial Interface in Client Mode

This section provides information about the function of the serial interface in client mode.

The Serial Interface (SI) client pins on the device are overlaid functions on the GPIO interface. SI client mode is enabled via strapping. For more information, see Section 3.4, Configuration Straps. When SI client mode is enabled, the appropriate GPIO pins are automatically overtaken. For more information, see Section 3.3, GPIO Alternate Functions.

Table 4-57 lists the registers associated with SI access client mode.

TABLE 4-57: SI ACCESS CLIENT MODE REGISTER

Register	Description
DEVCPU_GCB::IF_CTRL	Configuration of endianness and bit order.
DEVCPU_GCB::IF_CFGSTAT	Configuration of padding

The serial interface implements a SPI-compatible protocol that allows an external CPU to perform read and write access to register targets within the device. Endianness and bit order is configurable, and several options for high frequencies are supported.

TABLE 4-58: SI CLIENT MODE PINS

Pin Name	I/O	Description
SI_nCS	I	Active low chip select
SI_CLK	I	Clock input
SI_DI	I	Data input (MOSI)
SI_DO	O	Data output (MISO)

SI_DI is sampled on rising edge of SI_CLK. SI_DO is driven on falling edge of SI_CLK. There are no requirements on the logical values of the SI_CLK and SI_DI inputs when SI_nCS is asserted or deasserted; they can be either 0 or 1. SI_DO is only driven during read-access when read-data is shifted out of the device.

The external CPU initiates access by asserting chip select and then transmitting one bit read/write indication, 23 address bits, and 32 bits of write-data (or don't care bits when reading).

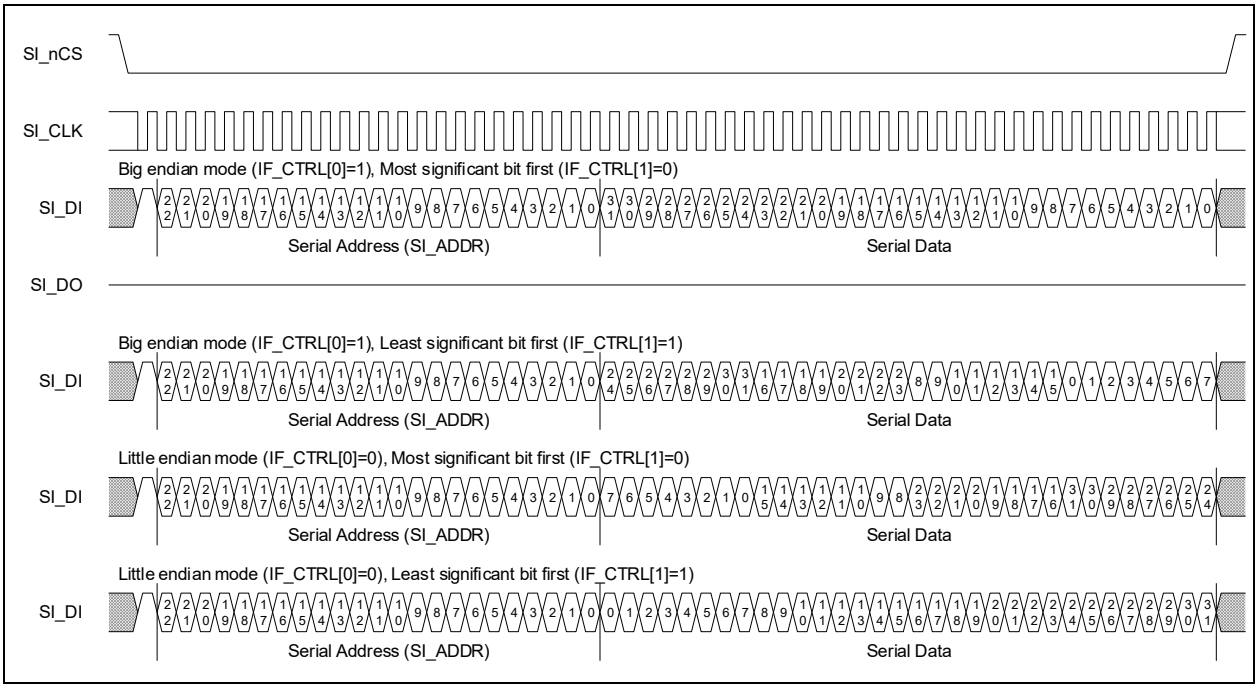
With the register address of a specific register (REG_ADDR), the SI address (SI_ADDR) is calculated as:

SI_ADDR = (REG_ADDR & 0x01FFFFFF)>>2

Data word endianness is configured through IF_CTRL[0]. The order of the data bits is configured using IF_CTRL[1].

Figure 4-47 shows various configurations for SI write access. The order of the data bits during writing, as depicted, is also used when the device is transmitting data during read operations.

FIGURE 4-47: WRITE SEQUENCE FOR SI



When reading registers using the SI interface, the device needs to prepare read data after receiving the last address bit. The access time of the register that is read must be satisfied before shifting out the first bit of read data. The external CPU must apply one of the following solutions to satisfy read access time.

- Use SI_CLK with a period of minimum twice the access time for the register target. For example, for normal switch core targets (single initiator):

  1/(2 × 400 ns) = 1.25 MHz (maximum)

- Pause the SI_CLK between shifting of serial address bit 0 and the first data bit with enough time to satisfy the access time for the register target.
- Configure the device to send out padding bytes before transmitting the read data to satisfy the access time for the register target. For example, 1 dummy byte allows enough read time for the SI clock to run up to 16 MHz in a single initiator system (see below for calculation).

The device is configured for inserting padding bytes by writing to IF_CFGSTAT.IF_CFG, these bytes are transmitted before the read data. The maximum frequency of the SI clock is calculated as:

(IF_CFGSTAT.IF_CFG × 8 - 1.5)/access time

For example, for normal switch core targets (single initiator), 1 byte padding gives (1 × 8 - 1.5)/400 ns = 16 MHz (maximum).

The maximum applicable SI clock frequency depends on the implementation of the external SI initiator, as well as the timing characteristics of the SI pins. The SI_DO output is kept tri-stated until the actual read data is transmitted.

Figure 4-48, Figure 4-49, and Figure 4-50 show options for serial read access. The illustrations show only one mapping of read data, the little endian with most significant bit first. Any of the mappings can be configured and applied to read data in the same way as for write data.

FIGURE 4-48: READ SEQUENCE FOR SI_CLK SLOW

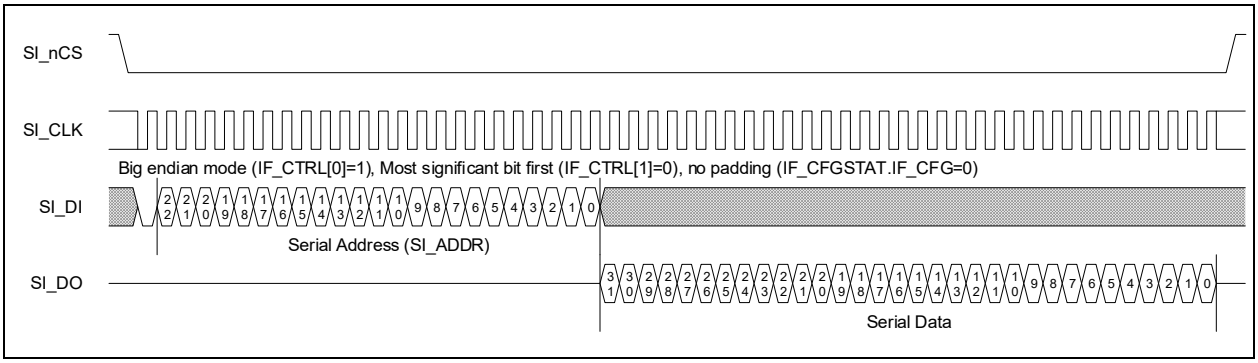


FIGURE 4-49: READ SEQUENCE FOR SI_CLK PAUSE

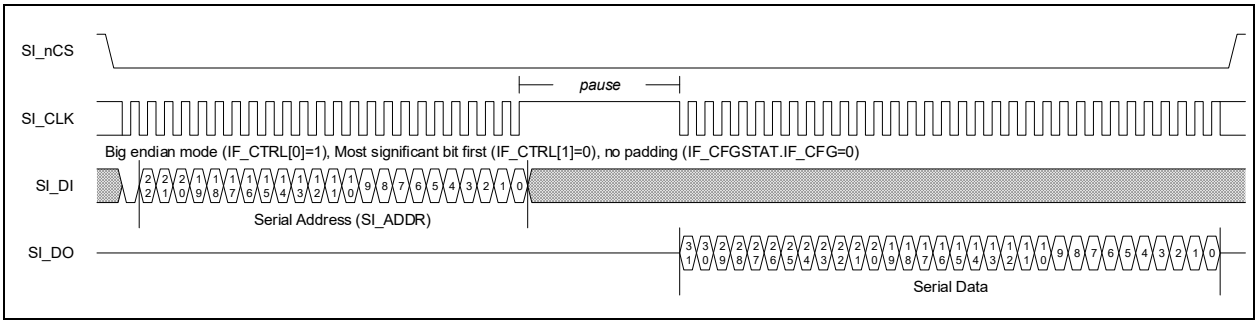
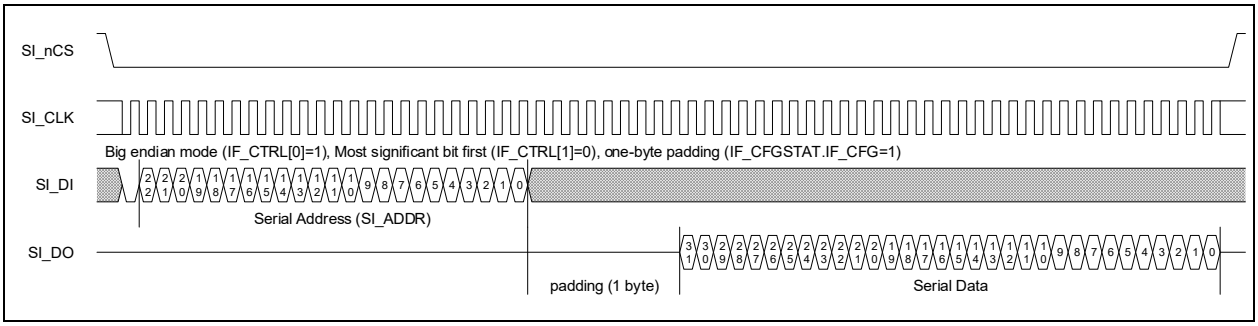


FIGURE 4-50: READ SEQUENCE FOR ONE-BYTE PADDING



When dummy bytes are enabled (IF_CFGSTAT.IF_CFG), the SI client logic enables an error check that sends out 0x88888888 and sets IF_CFGSTAT.IF_STAT if the SI controller does not provide enough time for register read.

When using SI, the external CPU must configure the IF_CTRL register after power-up, reset, or chip-level soft reset. The IF_CTRL register is constructed so that it can be written no matter the state of the interface.

4.23.1.2 MIIM Interface in Client Mode

This section provides the functional aspects of the MIIM client interface.

The MIIM client interface allows an external CPU to perform read and write access to the Switch Core register targets in the device. Register access is done indirectly, because the address and data fields of the MIIM protocol is less than those used by the register targets. Transfers on the MIIM interface are using the Management Frame Format protocol specified in IEEE 802.3, Clause 22.

The MIIM client pins on the device are overlaid functions on the GPIO interface. MIIM client mode is enabled via strapping. For more information, see Section 3.4, Configuration Straps. When MIIM client mode is enabled, the appropriate GPIO pins are automatically overtaken. For more information, see Section 3.3, GPIO Alternate Functions.

Table 4-59 lists the pins of the MIIM client interface.

TABLE 4-59: MIIM CLIENT PINS

Pin Name	I/O	Description
MIIM_SLV_MDC/GPIO	I	MIIM client clock input
MIIM_SLV_MDIO/GPIO	I/O	MIIM client t data input/output
MIIM_SLV_ADDR/GPIO	I	MIIM clientt address select

MIIM_SLV_MDIO is sampled or changed on the rising edge of MIIM_SLV_MDC by the MIIM client interface.

The MIIM client can be configured to answer on one of two different PHY addresses using the MIIM_SLV_ADDR pin. Setting the MIIM_SLV_ADDR pin to 0 configures the MIIM client to use PHY address 0, and setting it to 1 configures the MIIM client to use PHY address 31.

The MIIM client has seven 16-bit MIIM registers defined as listed in Table 4-60.

TABLE 4-60: MIIM CLIENT REGISTERS

Register Address	Register Name	Description
0	ADDR_REG0	Bits 15:0 of the address to read or write. The address field must be formatted as word address.
1	ADDR_REG1	Bits 31:16 of the address to read or write.
2	DATA_REG0	Bits 15:0 of the data to read or write. Returns 0x8888 if a register read error occurred.

TABLE 4-60: MIIM CLIENT REGISTERS (CONTINUED)

Register Address	Register Name	Description
3	DATA_REG1	Bits 31:16 of the data to read or write. The read or write operation is initiated after this register is read or written. Returns 0x8888 if read while busy or a register read error occurred.
4	DATA_REG1_INCR	Bits 31:16 of data to read or write. The read or write operation is initiated after this register is read or written. When the operation is complete, the address register is incremented by one. Returns 0x8888 if read while busy or a register read error occurred.
5	DATA_REG1_INERT	Bits 31:16 of data to read or write. Reading or writing to this register does not cause a register access to be initiated. Returns 0x8888 if a register read error occurred.
6	STAT_REG	The status register gives the status of any ongoing operations. Bit 0: Busy. Set while a register read/write operation is in progress. Bit 1: Busy_rd. Busy status during the last read or write operation. Bit 2: Err. Set if a register access error occurred. Others: Reserved.

A 32-bit switch core register read or write transaction over the MIIM interface is done indirectly due to the limited data width of the MIIM frame. First, the address of the register inside the device must be set in the two 16-bit address registers of the MIIM client using two MIIM write transactions. Afterwards the two 16-bit data registers can be read or written to access the data value of the register inside the device. Thus, it requires up to four MIIM transactions to perform a single read or write operation on a register target.

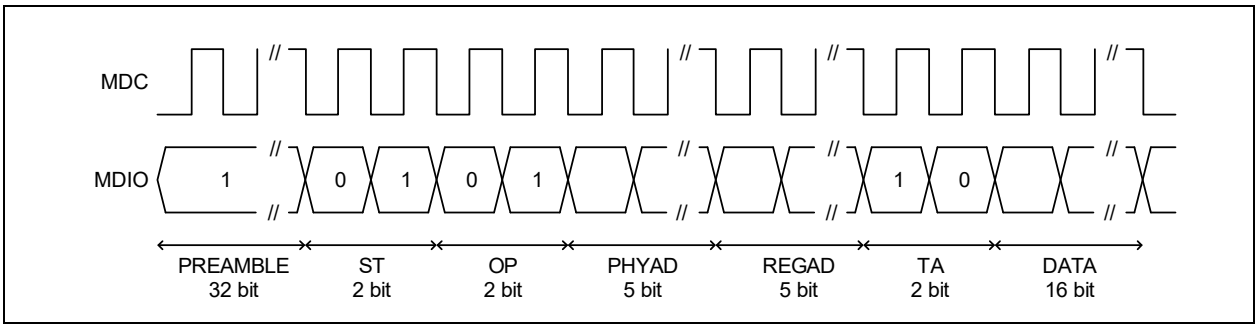
The address of the register to read/write is set in registers ADDR_REG0 and ADDR_REG1. The data to write to the register pointed to by the address in ADDR_REG0 and ADDR_REG1 is first written to DATA_REG0 and then to DATA_REG1. When the write transaction to DATA_REG1 is completed, the MIIM client initiates the switch core register write.

With the register address of a specific register (REG_ADDR), the MIIM address (MIIM_ADDR) is calculated as:

MIIM_ADDR = (REG_ADDR & 0x01FFFFFF)>>2

Figure 4-51 shows a single MIIM write transaction on the MIIM interface.

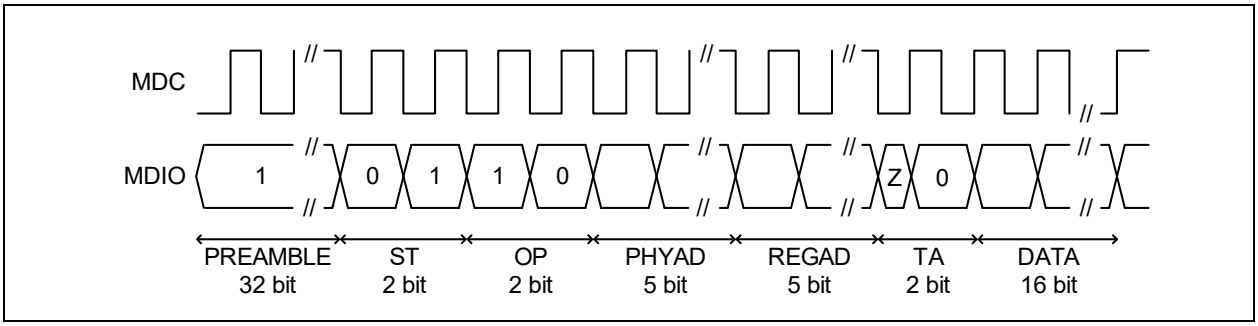
FIGURE 4-51: MIIM CLIENT WRITE SEQUENCE



A read transaction is done in a similar way. First, read the DATA_REG0 and then read the DATA_REG1. As with a write operation, the switch core register read is not initiated until the DATA_REG1 register is read. In other words, the returned read value is from the previous read transaction.

Figure 4-52 shows a single MIIM read transaction on the MIIM interface.

FIGURE 4-52: MIIM CLIENT READ SEQUENCE



4.23.1.3 I²C interface in Client Mode

This section provides the functional aspects of the I²C client interface.

The I²C client interface allows an external CPU to perform read and write access to 32-bit switch core registers in the device.

The I²C client pins on the device are overlaid functions on the GPIO interface. I²C client mode is enabled via strapping. For more information, see Section 3.4, Configuration Straps. When I²C client mode is enabled, the appropriate GPIO pins are automatically overtaken. For more information, see Section 3.3, GPIO Alternate Functions.

Table 4-61 lists the pins of the I²C client interface.

TABLE 4-61: I²C CLIENT PINS

Pin Name	I/O	Description
I2C_SLV_SCL/GPIO	I/O	I²C client clock input/output
I2C_SLV_SDA/GPIO	I/O	I²C client data input/output

I2C_SLV_SDA is sampled on the rising edge of I2C_SLV_SCL and driven on the falling edge of I2C_SLV_SCL. Clock stretching is supported. The register address used in the I²C frame is the register-indexed address (word-aligned address of the register right-shifted by 2 positions). The I²C client responds to the I²C address 95 (0x5F in hex).

A I²C frame for a register write is:

// Register write

// S D6~D0 W ACK A23~A16 ACK A15~A8 ACK A7~A0 ACK WD31~WD24 ACK WD23~WD16 ACK WD15~WD8 ACK WD7~WD0 ACK P

A I²C frame for a register read is:

// Register read

// S D6~D0 W ACK A23~A16 ACK A15~A8 ACK A7~A0 ACK S D6~D0 R ACK RD31~RD24 ACK RD23~RD16 ACK RD15~RD8 ACK RD7~RD0 NACK P

where:

D6-D0: The I²C address (must be 0x5F in order for the DUT to respond to the transaction)

A23-A0: The register address calculated as described above

WD31~WD0: The data to write to the register

RD31~RD0: The data read from the register

S: Start

P: Stop

W: Write

R: Read

4.23.2 FRAME INJECTION/EXTRACTION

The device support a mechanism for manual frame injection and extraction from an external CPU managed by software. The CPU ports of the switch core need to be configured (see Section 4.18, CPU Port Module). The mapping of the CPU ports to manual injection and extraction is done on the CPU port through DEVCPU_QS:XTR:XTR_GRP_CFG[0-1].MODE and DEVCPU_QS:INJ:INJ_GRP_CFG[0-1].MODE respectively.

4.23.2.1 Manual Injection Extraction

For more information on manual injection/extraction, refer to the field descriptions of the registers listed in Table 4-62.

TABLE 4-62: MANUAL INJECTION EXTRACTION REGISTER LIST

Register	Description
DEVCPU_QS:XTR:XTR_GRP_CFG[0-1]	Extraction group configuration
DEVCPU_QS:XTR:XTR_RD[0-1]	Manual extraction data
DEVCPU_QS:XTR:XTR_FRM_PRUNING[0-1]	Extraction frame pruning
DEVCPU_QS:XTR:XTR_FLUSH	Extraction group flush
DEVCPU_QS:XTR:XTR_DATA_PRESENT	Extraction status
DEVCPU_QS:INJ:INJ_GRP_CFG[0-1]	Injection group configuration
DEVCPU_QS:INJ:INJ_WR[0-1]	Manual injection data
DEVCPU_QS:INJ:INJ_CTRL[0-1]	Injection control
DEVCPU_QS:INJ:INJ_STATUS	Injection status
DEVCPU_QS:INJ:INJ_ERR[0-1]	Injection errors

4.23.3 VCORE SYSTEM PERIPHERALS

This section describes the sub-blocks of the VCore system.

4.23.3.1 MII Management Controller

This section provides information about the MII Management controllers. Table 4-63 lists the registers associated with the MII Management controllers.

TABLE 4-63: MIIM REGISTERS

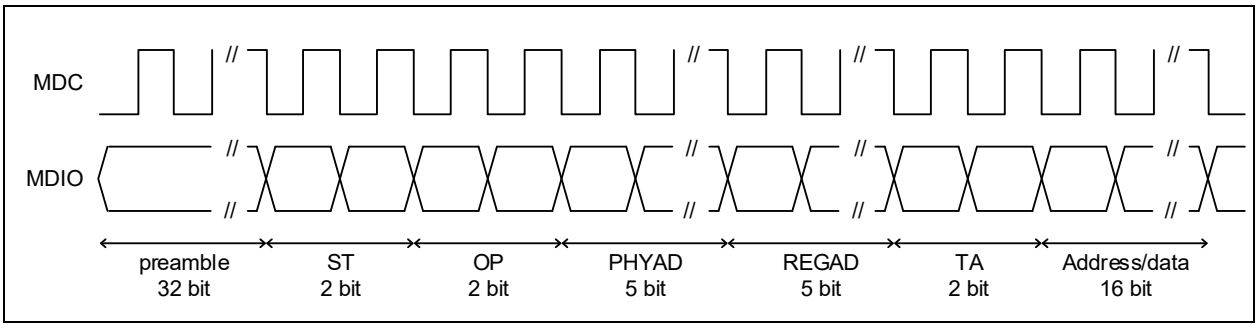
Register	Description
DEVCPU_GCB::MII_STATUS	Controller status
DEVCPU_GCB::MII_CMD	Command and write data
DEVCPU_GCB::MII_DATA	Read data
DEVCPU_GCB::MII_CFG	Clock frequency configuration
DEVCPU_GCB::MII_SCAN_0	Auto-scan address range
DEVCPU_GCB::MII_SCAN_1	Auto-scan mask and expects
DEVCPU_GCB::MII_SCAN_LAST_RSLTS	Auto-scan result
DEVCPU_GCB::MII_SCAN_LAST_RSLTS_VLD	Auto-scan result valid
DEVCPU_GCB:: MII_SCAN_RSLTS_STICKY	Differences in expected versus read auto-scan

The devices contain two MIIM controllers with equal functionality. Data is transferred on the MIIM interface using the Management Frame Format protocol specified in IEEE 802.3, Clause 22 or the MDIO Manageable Device protocol defined in IEEE 802.3, Clause 45. The Clause 45 protocol differs from the Clause 22 protocol by using indirect register access to increase the address range. The controller supports both Clause 22 and Clause 45.

The MIIM interface pins are overlaid functions on the GPIO interface. Before using these MIIM controllers, the overlaid functions for the appropriate GPIO pins must first be enabled. For more information, see Section 3.3, GPIO Alternate Functions. Refer to Table 3-9 for pin description information.

The MDIO signal is changed or sampled on the falling edge of the MDC clock by the controller. The MDIO pin is tri-stated in between access and when expecting read data.

FIGURE 4-53: MIIM MANAGEMENT TIMING



4.23.3.1.1 Clock Configuration

The frequency of the management interface clock generated by the MIIM controller is derived from the system frequency. The MIIM clock frequency is configurable and is selected with MII_CFG.MIIM_CFG_PRESCALE. The calculation of the resulting frequency is explained in the register description for MII_CFG.MIIM_CFG_PRESCALE.

4.23.3.1.2 MII Management PHY Access

Reads and writes across the MII management interface are performed through the MII_CMD register. Details of the operation, such as the PHY address, the register address of the PHY to be accessed, the operation to perform on the register (for example, read or write), and write data (for write operations), are set in the MII_CMD register. When the appropriate fields of MII_CMD are set, the operation is initiated by writing 0x1 to MII_CMD.MIIM_CMD_VLD. The register is automatically cleared when the MIIM command is initiated. When initiating single MIIM commands, MII_CMD.MIIM_CMD_SCAN must be set to 0x0.

When an operation is initiated, the current status of the operation can be read in MII_STATUS. The fields MII_STATUS.MIIM_STAT_PENDING_RD and MII_STATUS.MIIM_STAT_PENDING_WR can be used to poll for completion of the operation. For a read operation, the read data is available in MII_DATA.MIIM_DATA_RDDATA after completion of the operation. The value of MII_DATA.MIIM_DATA_RDDATA is only valid if MII_DATA.MIIM_DATA_SUCCESS indicates no read errors.

The MIIM controller contains a small command FIFO. Additional MIIM commands can be queued as long as MII_STATUS.MIIM_STAT_OPR_PEND is cleared. Care must be taken with read operations, because multiple queued read operations will overwrite MII_DATA.MIIM_DATA_RDDATA.

Note: A typical software implementation will never queue read operations, because the software needs read data before progressing the state of the software. In this case, MIIM_STATUS.MIIM_STAT_OPR_PEND is checked before issuing MIIM read or write commands, for read-operations MII_STATUS.MIIM_STAT_BUSY is checked before returning read result.

By default, the MIIM controller operates in Clause 22 mode. To access Clause 45 compatible PHYs, MII_CFG.MIIM_ST_CFG_FIELD and MII_CMD.MIIM_CMD_OPR_FIELD must be set according to Clause 45 mode of operation.

4.23.3.1.3 PHY Scanning

The MIIM controller can be configured to continuously read certain PHY registers and detect if the read value is different from an expected value. If a difference is detected, a special sticky bit register is set or a CPU interrupt is generated, or both. For example, the controller can be programmed to read the status registers of one or more PHYs and detect if the Link Status changed since the sticky register was last read.

The reading of the PHYs is performed sequentially with the low and high PHY numbers specified in MII_SCAN_0 as range bounds. The accessed address within each of the PHYs is specified in MII_CMD.MIIM_CMD_REGAD. The scanning begins when a 0x1 is written to MII_CMD.MIIM_CMD_SCAN and a read operation is specified in MII_CMD.MIIM_CMD_OPR_FIELD. Setting MII_CMD.MIIM_CMD_SINGLE_SCAN stops the scanning after all PHYs have been scanned one time. The remaining fields of MII_CMD register are not used when scanning is enabled.

The expected value for the PHY register is set in MII_SCAN_1.MIIM_SCAN_EXPECT. The expected value is compared to the read value after applying the mask set in MII_SCAN_1.MIIM_SCAN_MASK. To "don't care" a bit-position, write a 0 to the mask. If the expected value for a bit position differs from the read value during scanning, and the mask register has a 1 for the corresponding bit, a mismatch for the PHY is registered.

The scan results from the most recent scan can be read in MII_SCAN_LAST_RSLTS. The register contains one bit for each of the possible 32 PHYs. A mismatch during scanning is indicated by a 0. MII_SCAN_LAST_RSLTS_VLD will indicate for each PHY if the read operation performed during the scan was successful. The sticky-bit register MII_SCAN_RSLTS_STICKY has the mismatch bit set for all PHYs that had a mismatch during scanning since the last read of the sticky-bit register. When the register is read, its value is reset to all-ones (no mismatches).

4.23.3.2 Two-Wire Serial Interface

This section provides information about the TWI (two-wire serial interface) controllers. There are two independent TWI controller instances in the VCore System: I2C_0 and I2C_1.

The two-wire serial interface controller is compatible with the industry standard two-wire serial interface protocol. The controller supports standard speed up to 100 kbps and fast speed up to 400 kbps. Multiple bus controllers, as well as both 7-bit and 10-bit addressing, are also supported.

By default, the two-wire serial interface controller operates as a controller only.

The two-wire serial interface pins on the device are overlaid functions on the GPIO interface. Before enabling the two-wire serial interface, the overlaid functions for the appropriate GPIO pins must be enabled. For more information, see Section 3.3, GPIO Alternate Functions.

The following table lists the pins of the two-wire serial interface.

Table 4-64 lists the pins of the I^2C client interface.

TABLE 4-64: WIRE SERIAL INTERFACE PINS

Pin Name	I/O	Description
I2C0_SCL_*x*/GPIO	I/O	Multiplexed clock outputs, Open-collector output
I2C0_SDA/GPIO	I/O	Data, Open-collector output
I2C1_SCL/GPIO	I/O	Clock, Open-collector output
I2C1_SDA/GPIO	I/O	Data, Open-collector output

Setting IC_ENABLE.ENABLE enables the controller. The controller can be disabled by clearing the IC_ENABLE.ENABLE field, there is a chance that disabling is not allowed (at the time when it is attempted); the IC_ENABLE_STATUS register shows if the controller was successfully disabled.

The two-wire serial interface controller has an 8-byte combined receive and transmit FIFO.

4.23.3.2.1 Two-Wire Serial Interface Frequency Configuration

Before enabling the controller, the user must decide on either standard mode or fast mode (IC_CON.SPEED) and configure clock dividers for generating the correct timing (IC_SS_SCL_HCNT, IC_SS_SCL_LCNT, IC_FS_SCL_HCNT, IC_FS_SCL_LCNT). The configuration of the divider registers depends on the system clock frequency along with some constraints imposed by timing requirements.

The minimum timing requirements for TWI specify that the low time of the SCL clock must be at least 1300 ns, while the high time of the SCL clock must be at least 600 ns. The total SCL period (low plus high time) must be 2500 ns in fast mode and 10000 ns in standard mode.

TABLE 4-65: TWI MINIMUM TIMING REQUIREMENTS

Parameter	Value
SS_SCL_LOW	>= 1300 ns
SS_SCL_HIGH	>= 600 ns
SS_SCL	10000 ns
SS_SCL	SS_SCL_LOW + SS_SCL_HIGH
FS_SCL_LOW	>= 1300 ns

TABLE 4-65: TWI MINIMUM TIMING REQUIREMENTS (CONTINUED)

Parameter	Value
FS_SCL_HIGH	>= 600 ns
FS_SCL	2500 ns
FS_SCL	FS_SCL_LOW + FS_SCL_HIGH

Table 4-65 defines a range of applicable durations for FS_SCL_LOW, FS_SCL_HIGH, SS_SCL_LOW and SS_SCL_HIGH. The values for the configuration registers are computed as:

- IC_SS_SCL_LCNT = ceil(SS_SCL_LOW / clock period) - 1
- IC_SS_SCL_HCNT = ceil(SS_SCL_HIGH / clock period) - 8
- IC_FS_SCL_LCNT = ceil(FS_SCL_LOW / clock period) - 1
- IC_FS_SCL_HCNT = ceil(FS_SCL_HIGH / clock period) - 8

When IC_FS_SPKLEN is used for spike suppression, care must be taken to ensure that the following constraints are respected.

- IC_SS_SCL_LCNT > IC_FS_SPKLEN+7
- IC_FS_SCL_LCNT > IC_FS_SPKLEN+7
- IC_SS_SCL_HCNT > IC_FS_SPKLEN+5
- IC_FS_SCL_HCNT > IC_FS_SPKLEN+5

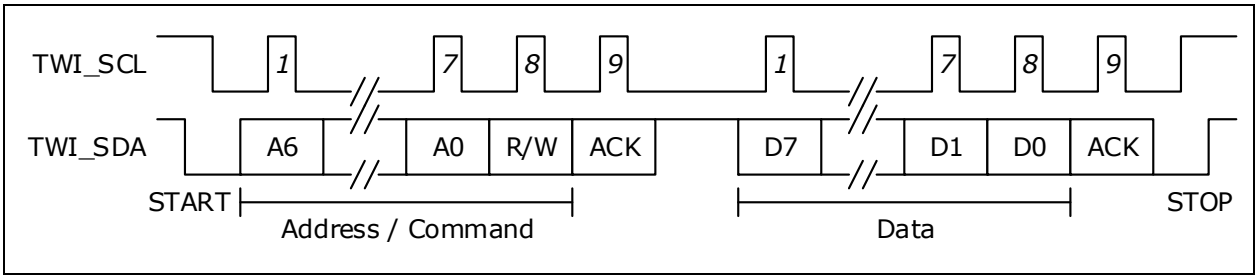
The Table 4-66 shows how the configuration registers affect the timing parameters of the two-wire serial interface. Depending on the requirements, the configuration registers should be trimmed accordingly to meet timing constraints.

TABLE 4-66: TWO-WIRE SERIAL INTERFACE TIMING PARAMETERS

Timing Parameter	Symbol	Standard Speed	Fast Speed
Setup time for a repeated START condition	tSU;STA	IC_SS_SCL_LCNT	IC_FS_SCL_HCNT
Hold time (repeated) START condition	tHD;STA	IC_SS_SCL_HCNT	IC_FS_SCL_HCNT
Setup time for STOP condition	tSU;STO	IC_SS_SCL_HCNT	IC_FS_SCL_HCNT
Bus free time between a STOP and a START condition	tBUF	IC_SS_SCL_LCNT	IC_FS_SCL_LCNT
Spike length	tSP	IC_FS_SPKLEN	IC_FS_SLKLEN
Data hold time	tHD;DAT	IC_SDA_HOLD	IC_SDA_HOLD
Data setup time	tSU;DAT	IC_SDA_SETUP	IC_SDA_SETUP

Some two-wire serial devices require a hold time on SDA after SCK when transmitting from the two-wire serial interface controller. The device supports a configurable hold delay through the IC_SDA_HOLD register.

FIGURE 4-54: TWO-WIRE SERIAL INTERFACE TIMING FOR 7-BIT ADDRESS ACCESS



During normal operation of the two-wire serial interface controller, the IC_STATUS register shows the activity and FIFO states.

4.23.3.2.2 Two-Wire Serial Interface Addressing

To configure either 7-bit or 10-bit addressing, use CFG.MASTER_10BITADDR.

The two-wire serial interface controller can generate both General Call and START Byte. Initiate this through IC_TAR.SPECIAL or TAR.GC_OR_START. The target address is configured using the TAR register.

4.23.3.2.3 Two-Wire Serial Interface Interrupt

The two-wire serial interface controller can generate a multitude of interrupts. All of these are described in the RAW_INTR_STAT register. The IC_RAW_INTR_STAT register contains interrupt fields that are always set when their trigger conditions occur. The IC_INTR_MASK register is used for masking interrupts and allowing interrupts to propagate to the IC_INTR_STAT register. When set in the IC_INTR_STAT register, the two-wire serial interface controller asserts an interrupt towards the VCore system.

The IC_RAW_INTR_STAT register also specifies what is required to clear the specific interrupts. When the source of the interrupt is removed, reading the appropriate IC_CLR_* register (for example, IC_CLR_RX_OVER) clears the interrupt.

4.23.3.2.4 Built-in Two-Wire Serial Multiplexer

The device has built-in support for connecting to multiple two-wire serial devices that use the same two-wire serial address. This is done by using the multiplexed clock outputs (TWI_SCL_GATEn) for the two-wire serial devices. Depending on which device or devices it needs to talk to, software can then enable/disable the various clocks. This multiplexing feature is available for I2C_0 controller.

From the two-wire serial controller's point of view, it does not know which TWI_SCL_GATEn clock is used. When using multiplexed mode, software needs to enable/disable individual clock connections before initiating the access operation using the two-wire serial controller. Feedback on the clock (from slow two-wire serial devices) is automatically done for the TWI_SCL_GATEn lines that are enabled.

To enable multiplexed clocks, first configure the TWI_SCL_GATEn overlaid mode in the GPIO controller. Individual clocks are then enabled by writing 1 to the corresponding GPIO output bit (in DEVCPU_GCB::GPIO_OUT). To disable the clock, write 0 to the GPIO output bit. Disabled clocks are not driven during access and the feedback is disabled.

Note: DEVCPU_GCB::GPIO_OUT_SET and DEVCPU_GCB::GPIO_OUT_CLR registers can be used to avoid read-modify-write of DEVCPU_GCB::GPIO_OUT.

4.23.3.3 GPIO Controller

This section provides information about the use of GPIO pins. Many of the GPIO pins have alternate functions. For more information, see Section 3.3, GPIO Alternate Functions.

Table 4-67 lists the registers associated with GPIO.

TABLE 4-67: GPIO REGISTERS

Register	Description
DEVCPU_GCB::GPIO_OUT	Value to drive on GPIO outputs
DEVCPU_GCB::GPIO_OUT_SET	Atomic set of bits in GPIO_OUT
DEVCPU_GCB::GPIO_OUT_CLR	Atomic clear of bits in GPIO_OUT
DEVCPU_GCB::GPIO_IN	Current value on the GPIO pins
DEVCPU_GCB::GPIO_OE	Enable of GPIO output mode (drive GPIOs)
DEVCPU_GCB::GPIO_ALT	Enable of overlaid GPIO functions
DEVCPU_GCB::GPIO_INTR	Interrupt on changed GPIO value
DEVCPU_GCB::GPIO_INTR_ENA	Enable interrupt on changed GPIO value
DEVCPU_GCB::GPIO_INTR_IDENT	Currently interrupting sources
DEVCPU_GCB::GPIO_SD_DEV_MAP	Mapping of parallel signal detects to ports
DEVCPU_GCB::GPIO_SD_SERDES_MAP	Mapping of parallel signal detects to SerDes

The GPIO pins are individually programmable. GPIOs are inputs by default and can be individually changed to outputs through GPIO_OE. The value of the GPIO pins is reflected in the GPIO_IN register. GPIOs that are in output mode are driven to the value specified in GPIO_OUT.

In a system where multiple different CPU threads (or different CPUs) may work on the GPIOs at the same time, the GPIO_OUT_SET and GPIO_OUT_CLR registers provide a way for each thread to safely control the output value of individual GPIOs, without having to implement locked regions and semaphores.

The GPIO_ALT registers are only reset by external reset to the device. This means that software reset of the DEVCPU_GCB is possible without affecting the mapping of overlaid functions on the GPIOs.

4.23.3.3.1 GPIO Alternate Function Configuration

Most of the GPIO pins have overlaid (alternative) functions that can be enabled through replicated GPIO_ALT registers.

To enable a particular GPIO pin index X with the alternative function, set the G_ALT register field in the replicated registers as follows:

TABLE 4-68: GPIO OVERLAID FUNCTIONS

ALT Mode	GPIO_ALTn[2].G_ALT[m]	GPIO_ALTn[1].G_ALT[m]	GPIO_ALTn[0].G_ALT[m]
0 (normal)	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Note: m = X% 32 (integer division remainder) and n = X/32 (integer division)

Example GPIO 17, ALT Mode 5

m = 17%32 = 17

n = 17/32 = 0 (when n is 0, then use empty string instead).

Therefore, the bit index 17 of the field G_ALT of registers GPIO_ALT[2], GPIO_ALT[1] and GPIO_ALT[0] has to be configured as:

GPIO_ALT[2].g_ALT[17] = 1

GPIO_ALT[1].g_ALT[17] = 0

GPIO_ALT[0].g_ALT[17] = 1

Example GPIO 35, ALT Mode 3

m = 35%32 = 3

n = 35/32 = 1

Therefore, the bit index 0 of the field G_ALT of registers GPIO_ALT2[2], GPIO_ALT2[1] and GPIO_ALT2[0] has to be configured as:

GPIO_ALT1[2].g_ALT[3] = 0

GPIO_ALT1[1].g_ALT[3] = 1

GPIO_ALT1[0].g_ALT[3] = 1

For more information, see Section 3.3, GPIO Alternate Functions.

4.23.3.3.2 GPIO Interface Mode

When the interface mode is enabled for a GPIO, then the mapping of the GPIO pin is not controlled by the GPIO_ALT registers anymore. Interface mode is typically enabled by hardware itself via strapping. Register TRL:IF_REGS:GENERAL_CTRL can be used to read and modify the current interface modes enabled.

4.23.3.3.3 GPIO Interrupt

The GPIO controller continually monitors all inputs and set bits in the GPIO_INTR register whenever a GPIO changes its input value. By enabling specific GPIO pins in the GPIO_INTR_ENA register, a change indication from GPIO_INTR is allowed to propagate (as GPIO interrupt) from the GPIO controller to the interrupt controller.

The currently interrupting sources can be read from GPIO_INTR_IDENT, this register is the result of a binary AND between the GPIO_INTR and GPIO_INTR_ENA registers.

4.23.3.3.4 Parallel Signal Detect

The GPIO controller has programmable parallel signal detects named SFPn_SD. When parallel signal detect is enabled for a front-port index or a SerDes index, it overrides the signal-detect/loss-of-signal value provided by the serial GPIO controller.

To enable parallel signal detect *n*, first configure which indices from the serial GPIO controller must be overwritten by setting GPIO_SD_DEV_MAP[*n*].G_SD_DEV_MAP and SGPIO_SD_SERDES_MAP[*n*].G_SD_SERDES_MAP. Then enable the SFP*n*_SD function on the GPIOs.

Table 4-69 lists parallel signal detect pins, which are overlaid on GPIOs. For more information, see Section 3.3, GPIO Alternate Functions.

TABLE 4-69: PARALLEL SIGNAL DETECT PINS

Register	I/O	Description
SFPn_SD/GPIO	I	Parallel signal detect n, where n is 0-1

4.23.3.4 Serial GPIO Controller

The device features one Serial GPIO (SIO) controller. By using a serial interface, the SIO controller significantly extends the number of available GPIOs with a minimum number of additional pins on the device. The primary purpose of the SIO controller is to connect control signals from SFP modules and to act as an LED controller.

The SIO controller supports up to 128 serial GPIOs (SGPIOs) organized into 32 ports, with four SGPIOs per port. Table 4-70 lists the registers associated with the SIO controller.

TABLE 4-70: SIO REGISTERS

Register	Description
DEVCPU_GCB::SIO_INPUT_DATA	Input data
DEVCPU_GCB::SIO_CFG	General configuration
DEVCPU_GCB::SIO_CLOCK	Clock configuration
DEVCPU_GCB::SIO_PORT_CFG	Output port configuration
DEVCPU_GCB::SIO_PORT_ENA	Port enable
DEVCPU_GCB::SIO_PWM_CFG	PWM configuration
DEVCPU_GCB::SIO_INTR_POL	Interrupt polarity
DEVCPU_GCB::SIO_INTR_RAW	Raw interrupt status
DEVCPU_GCB::SIO_INTR_TRIGGER	Interrupt trigger mode configuration
DEVCPU_GCB::SIO_INTR	Currently interrupting SGPIOs
DEVCPU_GCB::SIO_INTR_ENA	Interrupt enable
DEVCPU_GCB::SIO_INTR_IDENT	Currently active interrupts
DEVCPU_GCB::HW_SGPIO_TO_SD_MAP_CFG	Configure mapping of input SPGIOs to device signal detect
DEVCPU_GCB::HW_SPGIO_TO_SERDES_SD_MAP_CFG	Configure mapping of input SPGIOs to SerDes signal detect

SIO controller pin functions are overlaid on GPIO pins. For more information, see Section 3.3, GPIO Alternate Functions.

The SIO controller works by shifting SGPIO values out on SGn_DO through a chain of shift registers on the PCB. After shifting a configurable number of SGPIO bits, the SIO controller asserts SGn_LD, which causes the shift registers to apply the values of the shifted bits to outputs. The SIO controller can also read inputs while shifting out SGPIO values on SGn_DO by sampling the SGn_DI input. The values sampled on SGn_DI are made available to software.

If the SIO controller is only used for outputs, the use of the load signal is optional. If the load signal is omitted, simpler shift registers (without load) can be used, however, the outputs of these registers will toggle during shifting.

When driving LED outputs, it is acceptable that the outputs will toggle when SGPIO values are updated (shifted through the chain). When the shift frequency is fast, the human eye is not able to see the shifting though the LEDs.

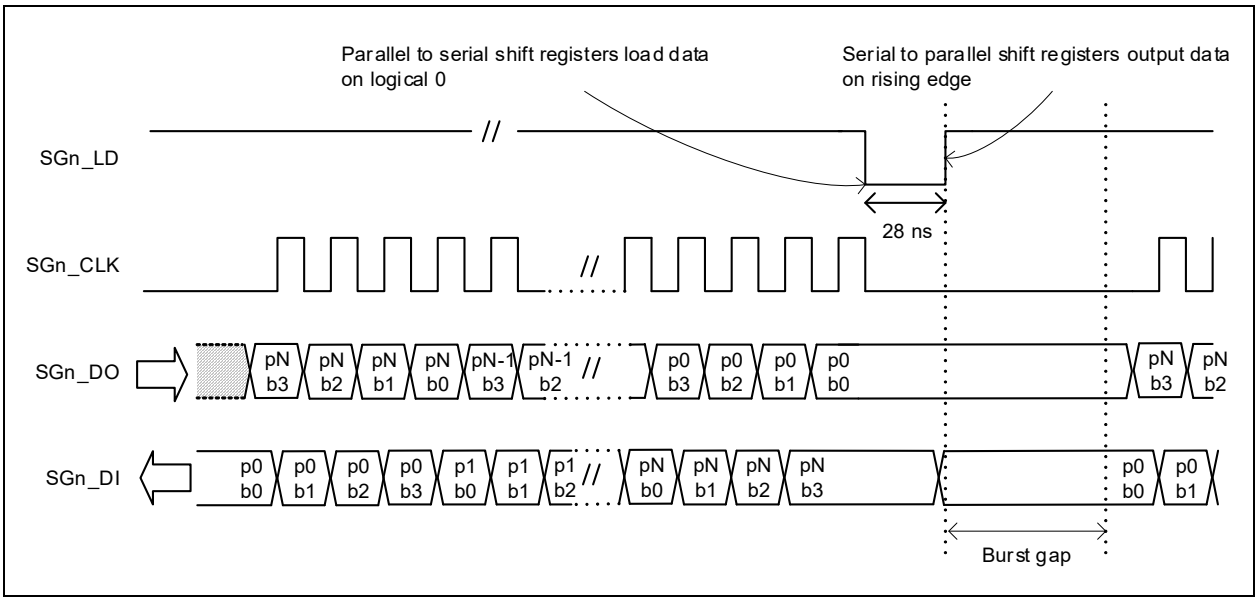
The number of shift registers in the chain is configurable. The SIO controller allows enabling of individual ports through SIO_PORT_ENA; only enabled ports are shifted out on SGn_DO. Ports that are not enabled are skipped during shifting of GPIO values.

Note: SIO_PORT_ENA allows skipping of ports in the SGPIO output stream that are not in use. The number of GPIOs per (enabled) port is configurable as well, through SIO_CFG.SIO_PORT_WIDTH this can be set to 1, 2, 3, or 4 bits. The number of bits per port is common for all enabled ports, so the number of shift registers on the PCB must be equal to the number of enabled ports times the number of SGPIOs per port.

Enabling of ports and configuration of SGPIO per port applies to both output mode and input mode. Unlike a regular GPIO port, a single SGPIO position can be used both as output and input. That is, software can control the output of the shift register AND read the input value at the same time. Using SGPIOs as inputs requires load-capable shift registers.

Regular shift registers and load-capable shift-registers can be mixed, which is useful when driving LED indications for integrated PHYs while supporting reading of link status from SFP modules.

FIGURE 4-55: SIO TIMING



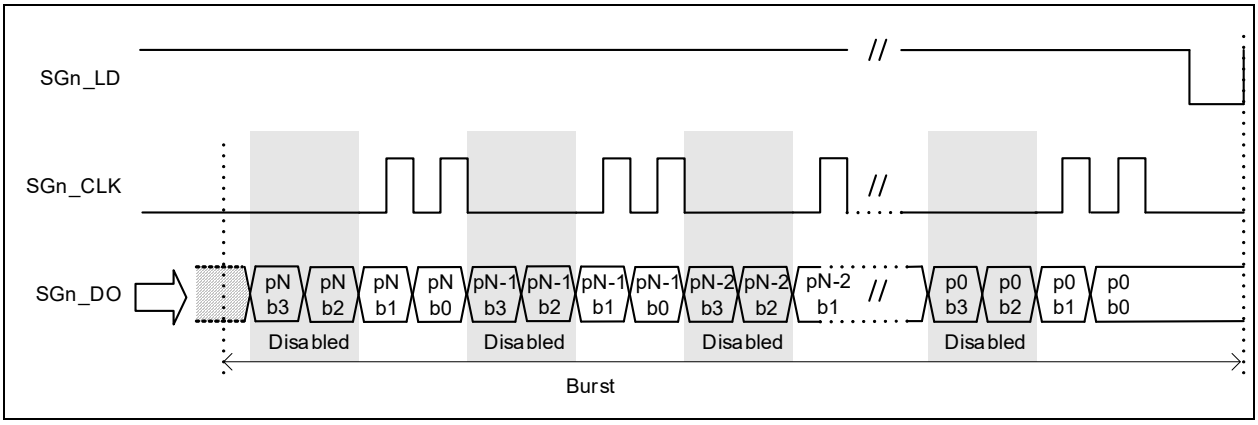
The SGPIO values are output in bursts followed by assertion of the SGn_LD signal. Values can be output as a single burst or as continuous bursts separated by a configurable burst gap. The maximum length of a burst is 32 × 4 data cycles. The burst gap is configurable in steps through SIO_CFG.SIO_BURST_GAP_DIS, SIO_CFG.SIO_BURST_GAP and SIO_CFG.SIO_REDUCED_GAP.

A single burst is issued by setting SIO_CFG.SIO_SINGLE_SHOT. The field is automatically cleared by hardware when the burst is finished. To issue continuous bursts, set SIO_CFG.SIO_AUTO_REPEAT. The SIO controller continues to issue bursts until SIO_CFG.SIO_AUTO_REPEAT is cleared.

SGPIO output values are configured in SIO_PORT_CFG.BIT_SOURCE. The input value is available in SIO_INPUT_-DATA.

Figure 4-56 shows what happens when the number of SGPIOs per port is configured to two (through SIO_CFG.SIO_PORT_WIDTH). Disabling of ports (through SIO_PORT_ENA) is handled in the same way as disabling the SGPIO ports.

FIGURE 4-56: SIO TIMING WITH SGPIOS DISABLED



The frequency of the SGn_CLK clock output is configured through SIO_CLOCK.SIO_CLK_FREQ. The SGn_LD output is asserted after each burst; this output is asserted for a period of 25 ns to 30 ns. The polarity of SGn_LD is configurable through SIO_CFG.SIO_LD_POLARITY.

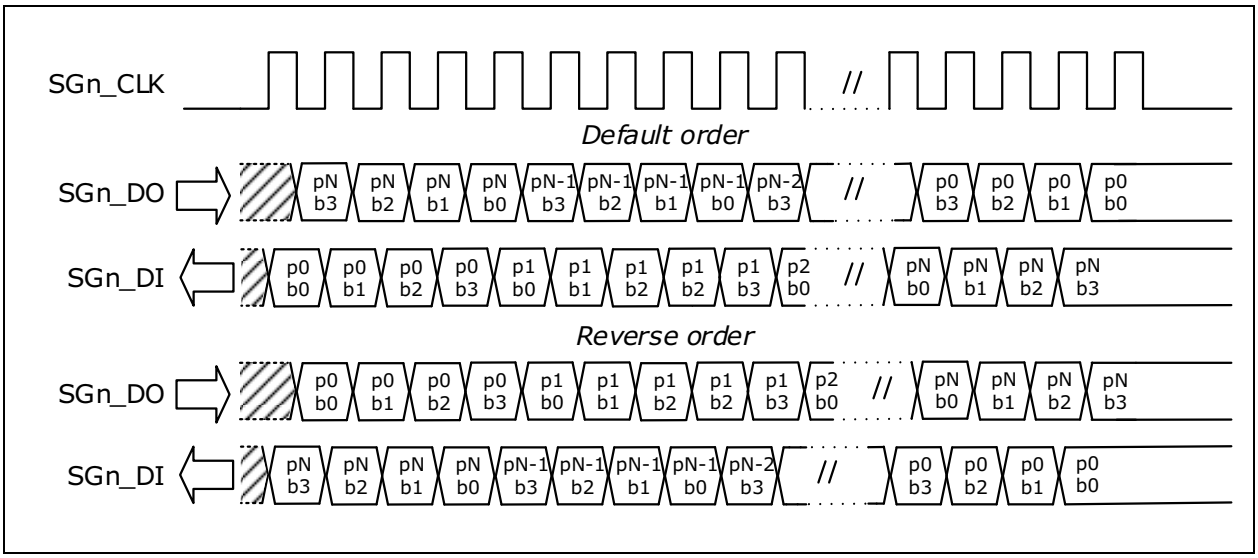
The SGn_LD output can be used to ensure that outputs are stable when serial data is being shifted through the registers. This can be done by using the SGn_LD output to shift the output values into serial-to-parallel registers after the burst is completed. If serial-to-parallel registers are not used, the outputs will toggle while the burst is being shifted through the chain of shift registers. A universal serial-to-parallel shift register outputs the data on a positive-edge load signal, and a universal parallel-to-serial shift register shifts data when the load pin is high, so one common load signal can be used for both input and output serial parallel conversion.

The assertion of SGn_LD happens after the burst to ensure that after power up, the single burst will result in well-defined output registers. Consequently, to sample input values one time, two consecutive bursts must be issued. The first burst results in the input values being sampled by the serial-to-parallel registers, and the second burst shifts the input values into the SIO controller.

The port order required in the serial bitstream depends on the physical layout of the shift register chain. Often the input and output port orders must be opposite in the serial streams. The port order of the input and output bitstream is independently configurable in SIO_CFG.SIO_REVERSE_INPUT and SIO_CFG.SIO_REVERSE_OUTPUT.

Figure 4-57 shows the port order.

FIGURE 4-57: SGPIO OUTPUT ORDER



4.23.3.4.1 Output Modes

The output mode of each SGPIO can be individually configured in SIO_PORT_CFG.BIT_SOURCE. The SIO controller features three output modes:

- Static
- Blink
- Link activity

The output mode can additionally be modified with PWM (SIO_PORT_CFG.PWM_SOURCE) and configurable polarity (SIO_PORT_CFG.BIT_POLARITY).

The Static mode is used to assign a fixed value to the SGPIO, for example, fixed 0 or fixed 1.

The Blink mode makes the SGPIO blink at a fixed rate. The SIO controller features two blink modes that can be set independently. A SGPIO can then be configured to use either blink mode 0 or blink mode 1. The blink outputs are configured in SIO_CFG.SIO_BMODE_0 and SIO_CFG.SIO_BMODE_1. To synchronize the blink modes between different devices, reset the blink counter using SIO_CFG.SIO_BLINK_RESET. All the SIO controllers on a device must be reset at same time to maintain the synchronization. The burst toggle mode of blink mode 1 toggles the output with every burst.

TABLE 4-71: BLINK MODES

Register	Description
Blink Mode 0	0: 5 Hz blink frequency 1: 2.5 Hz blink frequency 2: 1.25 Hz blink frequency 3: 0.625 Hz blink frequency
Blink Mode 1	0: 20 Hz blink frequency 1: 10 Hz blink frequency 2: 5 Hz blink frequency 3: Burst toggle

The Link Activity mode makes the output blink when there is activity on the port module (Rx or Tx). The mapping between SIO port number and device is listed in Table 4-72.

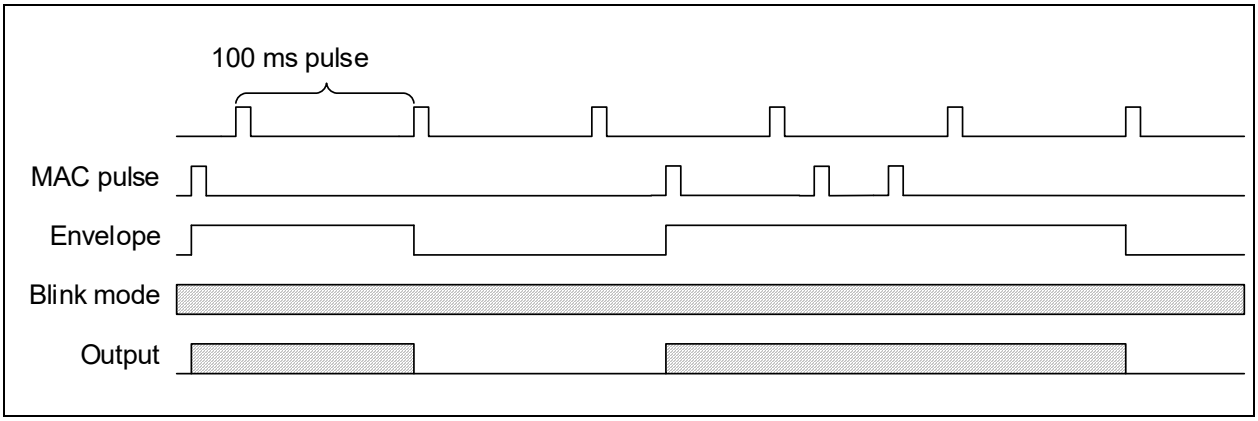
TABLE 4-72: SIO CONTROLLER PORT MAPPING

Register	Description
0	Device 0
1	Device 1
2	Device 2
3	Device 3
4	Device 4
5	Device 5
6	Device 6
7	Device 7

The Link Activity mode uses an envelope signal to gate the selected blinking pattern (blink mode 0 or blink mode 1). When the envelope signal is asserted, the output blinks, and when the envelope pattern is deasserted, the output is turned off. To ensure that even a single packet makes a visual blink, an activity pulse from the port module is extended to minimum 100 ms. If another packet is sent while the envelope signal is asserted, the activity pulse is extended by another 100 ms. The polarity of the link activity modes can be set in SIO_PORT_CFG.BIT_SOURCE.

Figure 4-58 shows the link activity timing.

FIGURE 4-58: LINK ACTIVITY TIMING



4.23.3.4.2 SIO Interrupt

The SIO controller can generate interrupts based on the value of the input value of the SGPIOs. All interrupts are level sensitive.

The SIO controller has interrupt registers that each has one bit for each of the 128 GPIOs:

1. Set the respective bit of SIO_INTR_ENA high to enable a GPIO interrupt.
2. SIO_INTR_RAW is high if the corresponding input bit is high (corrected for polarity as configured in SIO_INTR_POL). This register changes when the input changes.
3. SIO_INTR is high if the condition of the configured trigger mode for the bit is met. The trigger mode can be configured in SIO_INTR_TRIGGER0 and SIO_INTR_TRIGGER1 between level-triggered, edge-triggered, falling-edge-triggered, and rising-edge-triggered interrupt. This register is a sticky bit vector and can only be cleared by software. A bit is cleared by writing a 1 to the bit position.
4. SIO_INTR_IDENT is the result of SIO_INTR with the disabled interrupts (from SIO_INTR_ENA) removed. This register changes when SIO_INTR or the enable registers change.

The SIO controller has one interrupt output connected to the main interrupt controller, which is asserted when one or more interrupts in SIO_INTR_IDENT are active. To determine which SGPIO is causing the interrupt, the CPU must read this register. The interrupt output remains high until all interrupts in SIO_INTR_IDENT are cleared (either by clearing SIO_INTR or disabling the interrupts in SIO_INTR_ENA).

4.23.3.4.3 Loss of Signal Detection

The SIO controller can propagate loss of signal detection inputs directly to the external signal detection input of the port and SerDes modules. This is useful when, for example, SFP modules are connected to the device. Any SGPIO input can be mapped to any device or SerDes loss of signal input through HW_SGPIO_TO_SD_MAP_CFG and HW_SGPIO_TO_SERDES_SD_MAP_CFG respectively, by writing the global SGPIO index of the selected SGPIO to these registers. As mentioned in Section 4.23.3.4, Serial GPIO Controller, the SIO controller supports 128 serial GPIOs (SGPIOs) organized into 32 ports, with four SGPIOs per port. The global SGPIO index is calculated as:

SGPIO global index = (SGPIO port) * 4 + (SGPIO index within SGPIO port)

Loss of signal can also be taken directly from overlaid functions on the regular GPIOs. In that case, the input from the SIO controller is ignored. For more information, see Section 4.23.3.3.4, Parallel Signal Detect.

The polarity of the loss of signal input is configured using SIO_INT_POL, meaning the same polarity must be used for loss of signal detect and interrupt.

4.23.3.5 Temperature Sensor

This section provides information about the on-die temperature sensor. When enabled the temperature sensor logic will continually monitor the temperature of the die and make this available for software.

Table 4-73 lists the registers associated with the temperature monitor.

TABLE 4-73: TEMPERATURE SENSOR REGISTERS

Register	Description
CHIP_TOP:PVT_SENSOR:PVT_SENSOR_CFG	Enabling of sensor
CHIP_TOP:PVT_SENSOR:PVT_SENSOR_STAT	Temperature value

The temperature sensor is enabled by setting PVT_SENSOR_CFG.SAMPLE_ENA. After this the temperature sensor will sample temperature every 16 ms and show current temperature via PVT_SENSOR_STAT.DATA.

The PVT_SENSOR_CFG.PVT_CLK_CFG register must be configured to match the system clock frequency. The formula for converting DATA field value to centigrade temperature is:

Temp(C) = (-3.4627E-11)*N^4 + (1.1023E-07)*N^3 + (-1.9165E-04)*N^2 + (3.0604E-01)*N + (-5.6197E+01)

N = Value of CHIP_TOP:PVT_SENSOR:PVT_SENSOR_STAT.DATA

It will take approximately 16 ms after setting SAMPLE_ENA until the first temperature sample is ready. The PVT_ SENSOR_STAT.DATA_VALID field will be set when the temperature value is available.

4.23.3.6 Memory Integrity Monitor

Soft errors happen in all integrated circuits, these are a result of natural alpha decay, cosmic radiation, or electrical disturbances in the environment in which the device operates. The chance of soft-errors happening in a memory (RAM) is higher than for flip flop based logic, because the memory structures are physically small and changes require less outside force than in flip flops. The LAN9645xF provides built-in protection from soft errors by using error correcting code (ECC) on critical memories. In addition, the LAN9645xF allows monitoring and reporting of soft-error events.

Table 4-74 lists the registers associated with the memory integrity monitor.

TABLE 4-74: INTEGRITY MONITOR REGISTERS

Register	Description
DEVCPU_GCB::MEMITGR_CTRL	Trigger monitor state changes
DEVCPU_GCB::MEMITGR_STAT	Current state of the monitor and memory status
DEVCPU_GCB::MEMITGR_INFO	Shows indication when in DETECT state
DEVCPU_GCB::MEMITGR_IDX	Shows memory index when in DETECT state
DEVCPU_GCB::MEMITGR_DIV	Monitor speed

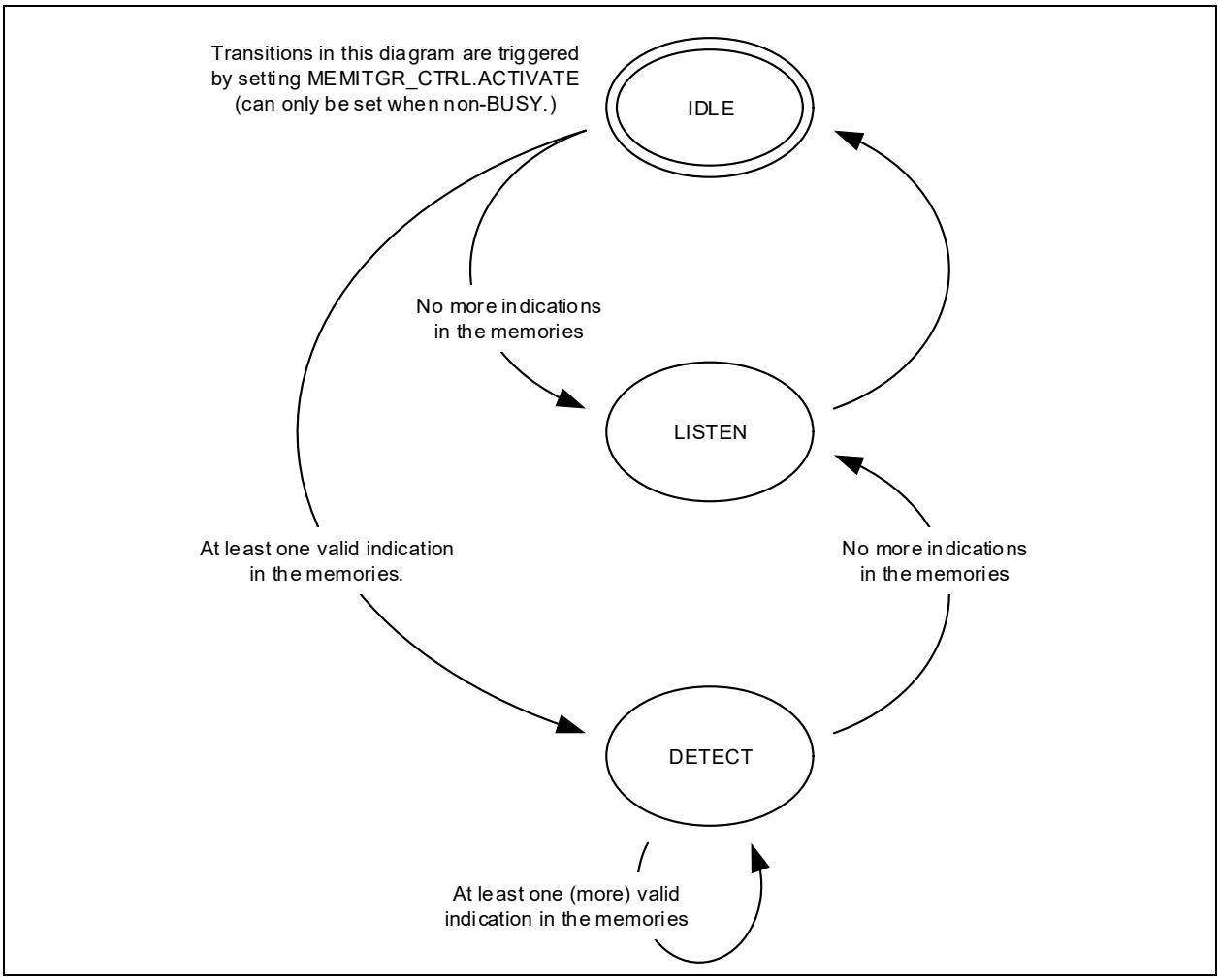
The memory integrity monitor looks for memory soft-error indications. Correctable (single bit) and non-correctable (multibit or parity) indications are detected during memory read and can be reported to software via "ITGR" software interrupt.

The memory integrity monitor operates in three different states: IDLE, LISTEN, and DETECT. After a reset, the monitor starts in the IDLE state.

- **IDLE**: The monitor is deactivated and in quiet mode. In IDLE mode, the memories still correct, detect, and store indications locally, but they are not able to report indications to the monitor.
- **LISTEN**: In the LISTEN state, the monitor looks for indications in the memories. MEMITGR_STAT.INDICATION is set (and interrupt is asserted) when indications are detected.
- **DETECT**: DETECT state is used when indications are read from the memories. It means that a valid indication is available in MEMITGR_INFO and the corresponding memory index in MEMITGR_IDX.

The current state of the monitor is reported in MEMITGR_STAT.MODE_IDLE, MEMITGR_STAT.MODE_DETECT, and MEMITGR_STAT.MODE_LISTEN. Software initiates transitions between states by setting the one-shot MEMITGR_CTRL.ACTIVATE field. It may take some time to transition from one state to the next. The MEMITGR_CTR.ACTIVATE field is not cleared before the next state is reached (also the MEMITGR_STAT.MODE_BUSY field is set while transitioning between states).

FIGURE 4-59: MONITOR STATE DIAGRAM



The first time after reset that MEMITGR_CTRL.ACTIVATE is set, the monitor resets the detection logic in all the memories and transitions directly from IDLE to LISTEN state.

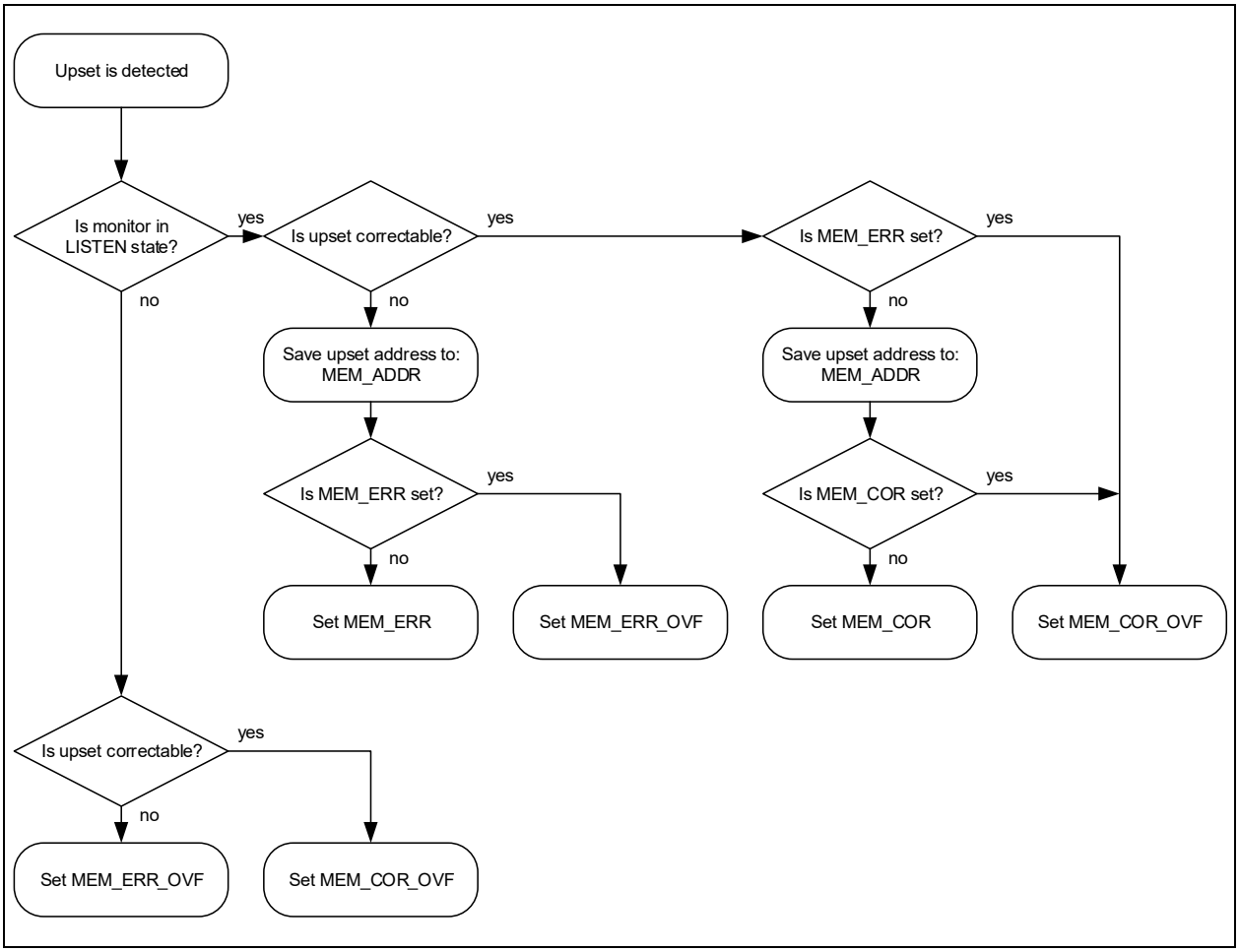
Before setting MEMITGR_CTRI.ACTIVATE for the first time, speed up the monitor by setting MEMITGR_DIV.MEM_DIV to the value specified in the registers list.

To read out indications, first transition from LISTEN to IDLE, then continue transitioning until the LISTEN state is reached. Every time the monitor ends up in the DETECT state, an indication is available in the MEMITGR_INFO and MEMITGR_IDX registers. Each memory stores one indication. Indications are cleared when they are read by way of the monitor. Each indication contains four flags and one memory address.

- The MEM_ERR flag is set when a non-correctable upset is detected and the corresponding address is available in MEM_ADDR.
- The MEM_ERR_OVF flag is set when a non-correctable upset is detected for which address could not be stored.
- The MEM_COR flag is set when a correctable upset is detected and the corresponding address is available in MEM_ADDR.
- The MEM_COR_OVF flag is set when a correctable upset is detected for which address could not be stored.

Information about non-correctable upsets is prioritized over correctable upsets. Address can only be saved when the monitor is in LISTEN mode. Figure 4-60 shows how the detection logic sets flags and address.

FIGURE 4-60: MEMORY DETECTION LOGIC



If the MEM_ERR_OVF or MEM_COR_OVF flag is set, at least one event has occurred for which the address could not be stored.

Table 4-75 shows ECC enabled memories in the device, their index, and the recommended approach for handling indications. If the controller reports an index that is not mentioned in the list, the recommended approach is to reboot the device.

TABLE 4-75: MEMORIES WITH INTEGRITY REPORT

Index	Description
Other	For unlisted indexes, the recommended approach is to reboot the device.

Reading from uninitialized memory locations has a high plausibility of triggering non-correctable or correctable indications. This is useful when developing integrity monitor software driver. For example, powering up a system without initializing the VCAPs and reading actions and sticky bits will trigger monitor indications. Note that the contents of memories are not changed by device reset, so power cycle is needed to reset the memories.

4.23.3.7 Interrupt Controller

This section provides information about the outbound VCore interrupt controller. Table 4-76 lists the registers associated with this interrupt controller.

TABLE 4-76: INTERRUPT CONTROLLER REGISTERS

Register	Description
INTR:INTR_RAW	Current value of interrupt inputs
INTR:INTR_BYPASS	Force non-sticky function
INTR:INTR_TRIGGER	Configure edge or level sensitive events
INTR:INTR_FORCE	Force events (for software debug)
INTR:INTR_STICKY	Currently logged events
INTR:INTR_ENA	Enable of interrupt sources
INTR:INTR_ENA_CLR	Atomic clear of bits in INTR_ENA
INTR:INTR_ENA_SET	Atomic set of bits in INTR_ENA
INTR:INTR_IDENT	Currently enabled and interrupting sources
INTR:DST_INTR_MAP	Mapping of interrupt sources to destinations
INTR:DST_INTR_IDENT	Currently enabled, mapped, and interrupting sources per destination
INTR:DEV_INTR_POL	Polarity of module interrupt inputs
INTR:DEV_INTR_RAW	Current value of module interrupts
INTR:DEV_INTR_BYPASS	Force non-sticky function for module interrupts
INTR:DEV_INTR_TRIGGER	Configure edge or level sensitive events for module interrupts
INTR:DEV_INTR_STICKY	Currently logged module interrupt events
INTR:DEV_INTR_ENA	Enable of module interrupts
INTR:DEV_INTR_IDENT	Currently interrupting and enabled module interrupts
INTR:EXT_SRC_INTR_POL	Polarity of external interrupt inputs.
INTR:EXT_DST_INTR_POL	Polarity of external interrupt outputs.
INTR:EXT_INTR_DRV	Drive mode for external interrupt outputs

The interrupt controller maps interrupt sources from VCore and switch-core blocks, port modules, and external interrupt inputs to a number of interrupt destinations, which can be transmitted from the LAN9645xF using the overlaid functions on GPIOs.

Table 4-77 lists the available interrupt sources in the LAN9645xF. The index can be used to access the appropriate bits in INTR:INTR_* and INTR:DST_INTR_* registers.

TABLE 4-77: INTERRUPT SOURCES

Index	Source Name	Description
0	DEV_ALL	Aggregated port module interrupt, this interrupt is asserted if there is an active and enabled interrupt from any of the port modules.

TABLE 4-77: INTERRUPT SOURCES (CONTINUED)

Index	Source Name	Description
1	RESERVED	
2	RESERVED	
3	RESERVED	
4	RESERVED	
5	RESERVED	
6	RESERVED	
7	RESERVED	
8	RESERVED	
9	TWI0	TWI interrupt.
10	TWI1	TWI interrupt.
11	RESERVED	
12	CUPHY0	Embedded CuPHY port 0 interrupt
13	CUPHY1	Embedded CuPHY port 1 interrupt
14	CUPHY2	Embedded CuPHY port 2 interrupt
15	CUPHY3	Embedded CuPHY port 3 interrupt
16	CUPHY4	Embedded CuPHY port 4 interrupt
17	ANA	Analyzer interrupt.
18	GPIO	Parallel GPIO interrupt.
19	RESERVED	
20	RESERVED	
21	MIIM0	MIIM controller 0 interrupt.
22	MIIM1	MIIM controller 1 interrupt.
23	XTR	Extraction data ready interrupt (manual extraction).
24	INJ	Injection ready interrupt (manual injection).
25	SGPIO	Serial GPIO interrupt.
26	PTP_SYNC	PTP synchronization interrupt.
27	PTP_BLK	Timestamp ready interrupt.
28	ITGR	Memory integrity interrupt.
29	SEMA0	DEVCPU_ORG[0]:DEVCPU_ORG:SEMA[0-7]
30	SEMA1	DEVCPU_ORG[0]:DEVCPU_ORG:SEMA[0-7]
31	SEMA2	DEVCPU_ORG[0]:DEVCPU_ORG:SEMA[0-7]
32	SEMA3	DEVCPU_ORG[0]:DEVCPU_ORG:SEMA[0-7]
33	SEMA4	DEVCPU_ORG[0]:DEVCPU_ORG:SEMA[0-7]
34	SEMA5	DEVCPU_ORG[0]:DEVCPU_ORG:SEMA[0-7]
35	SEMA6	DEVCPU_ORG[0]:DEVCPU_ORG:SEMA[0-7]
36	SEMA7	DEVCPU_ORG[0]:DEVCPU_ORG:SEMA[0-7]
37	UVOV	Under voltage over voltage controller interrupt
38	RESERVED	

Table 4-78 lists the available interrupt destinations in the device.

TABLE 4-78: INTERRUPT DESTINATIONS

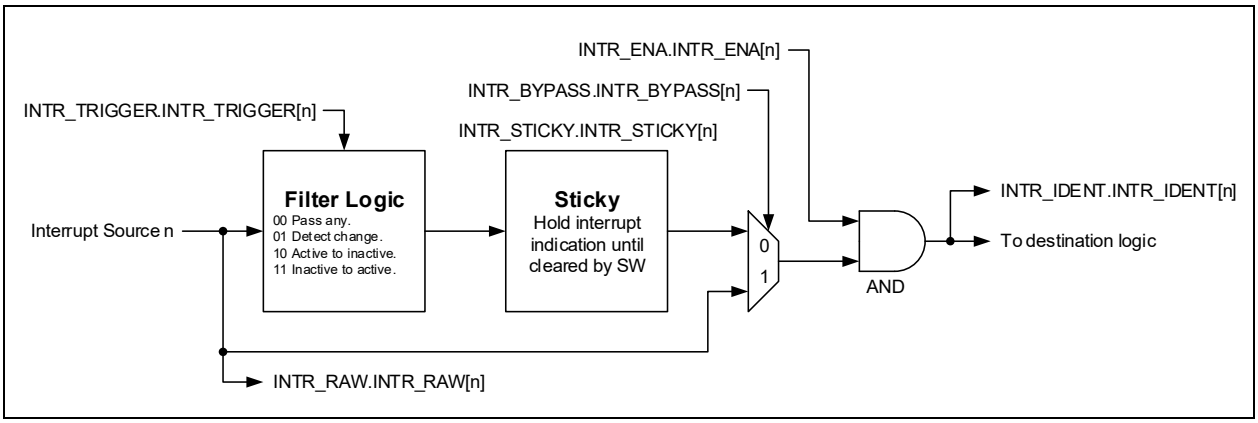
Destination Name	Description
EXT_DST0-5	External interrupt destination to IO

All interrupts, events, and indications inside in the interrupt controller are active high. If an interrupt source supports polarity correction, it is applied before going into the interrupt controller. If an interrupt destination supports polarity correction, it is applied after leaving the interrupt controller.

4.23.3.7.1 Interrupt Source Configuration

Interrupt sources are handled identically inside the interrupt controller. This section describes interrupt source n, which refers to the bit index of that interrupt source in the INTR_* and DST_INTR_* registers. Figure 4-61 shows the logic associated with a single interrupt source.

FIGURE 4-61: INTERRUPT SOURCE LOGIC



The current value of an interrupt source is available in INTR_RAW.INTR_RAW[n]. INTR_STICKY.INTR_STICKY[n] is set when the interrupt controller detects an interrupt. There are four detection methods:

- When INTR_TRIGGER.INTR_TRIGGER[n] is set to level-activated, the interrupt controller continually sets INTR_STICKY.INTR_STICKY[n] for as long as the interrupt source is active.
- When INTR_TRIGGER.INTR_TRIGGER[n] is set to edge-triggered, the interrupt controller only sets INTR_STICKY.INTR_STICKY[n] when the interrupt source changes value.
- When INTR_TRIGGER.INTR_TRIGGER[n] is set to falling-edge-triggered, the interrupt controller only sets INTR_STICKY.INTR_STICKY[n] when the interrupt source changes from active to inactive value.
- When INTR_TRIGGER.INTR_TRIGGER[n] is set to rising-edge-triggered, the interrupt controller only sets INTR_STICKY.INTR_STICKY[n] when the interrupt source changes from inactive to active value.

Software can clear INTR_STICKY.INTR_STICKY[n] by writing 1 to bit n. However, the interrupt controller will immediately set this bit again if the source input is still active (when INTR_TRIGGER is 0) or if it sees a triggering event on the source input (when INTR_TRIGGER different from 0).

The interrupt source is enabled in INTR_ENA.INTR_ENA[n]. When INTR_STICKY.INTR_STICKY[n] is set and the interrupt is enabled, the interrupt is indicated towards the interrupt destinations. An active and enabled interrupt source sets INTR_IDENT.INTR_IDENT[n].

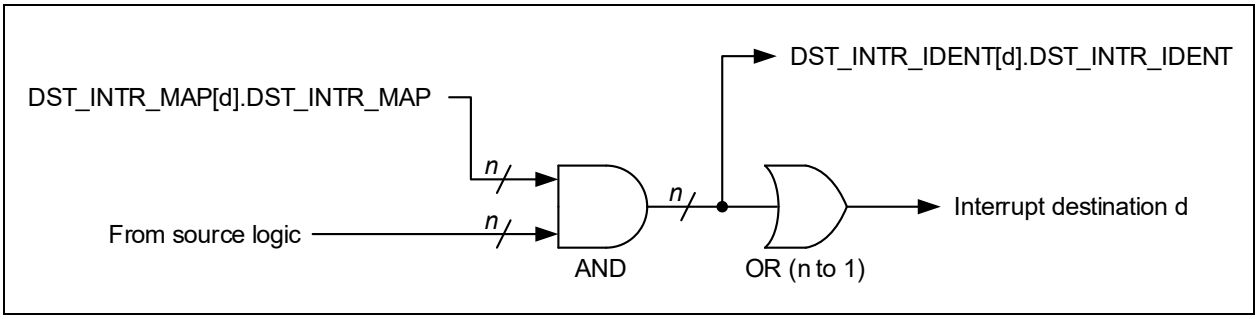
On rare occasions it is desirable to bypass the stickiness of interrupt sources and use INTR_RAW.INTR_RAW[n] directly instead of INTR_STICKY.INTR_STICKY[n]. Set INTR_BYPASS.INTR_BYPASS[n] to enable bypass and ignore INTR_STICKY and INTR_TRIGGER configurations.

Note: The bypass function may be useful for some software interrupt handler architectures. It should only be used for interrupt sources that are guaranteed to be sticky in the source block. For example, the GPIO interrupts that are generated from sticky bits in DEVCPU_GCB::GPIO_INTR may be applicable for the bypass mode.

4.23.3.7.2 Interrupt Destination Configuration

The nine interrupt destinations are handled identically in the interrupt controller. This section describes destination *d*, which refers to the replication index of that interrupt in the DST_INTR_* registers. Figure 4-62 shows the logic associated with a single interrupt destination.

FIGURE 4-62: INTERRUPT DESTINATION LOGIC



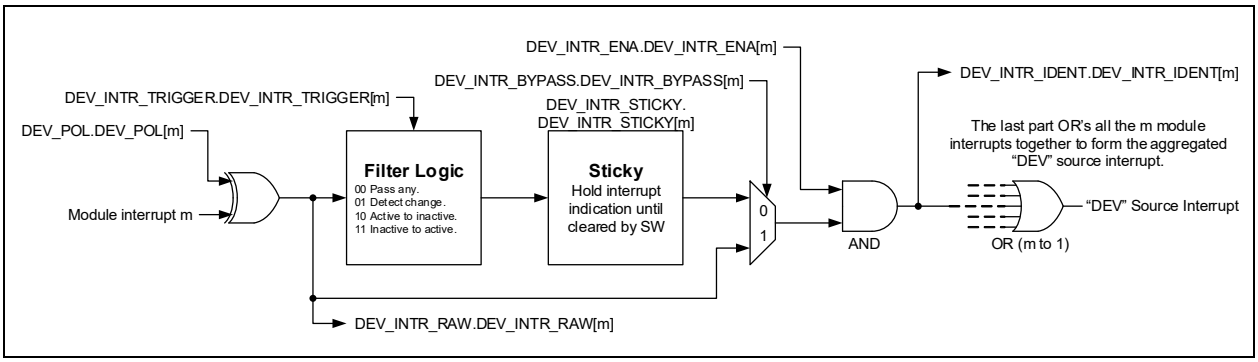
The interrupt destination can enable individual sources for interrupt by writing a mask to DST_INTR_MAP[*d*].DST_INTR_MAP. When a source is enabled in DST_INTR_MAP then an interrupt from this source will be propagated to the interrupt destination.

The currently active and enabled source interrupts for a destination can be seen by reading DST_INTR_IDENT[*d*].DST_INTR_IDENT.

4.23.3.7.3 Port Module Interrupts

Each port module can generate an interrupt. Because there are many modules to handle the interrupts in parallel with the other source interrupts in the INTR_* registers, the port module interrupts are aggregated in a separate source interrupt hierarchy before being presented to the interrupt controller source logic as the DEV source interrupt.

FIGURE 4-63: PORT MODULE INTERRUPT LOGIC



The module interrupt polarity is configurable in DEV_INTR_POL.DEV_INTR_POL[m].

DEV_INTR_RAW, DEV_INTR_TRIGGER, DEV_INTR_STICKY, DEV_INTR_BYPASS, DEV_INTR_ENA, and DEV_INTR_IDENT works in the same way as the INTR_* counterparts.

The final step when handling module interrupts is an aggregation of all individual module interrupts to the DEV source interrupt.

4.23.3.7.4 GPIO Mapped Interrupts

The interrupt controller supports a number of external GPIO mapped source interrupts (inputs to the device) and external GPIO destination interrupts (outputs from the device). The external interrupts are mapped to GPIOs using overlaid functions. For more information, see Section 3.3, GPIO Alternate Functions.

Source and destination interrupts work independently from each other and they can be used at the same time. The polarity (active high or low) of source and destination interrupts is configured in EXT_SRC_INTR_POL and EXT_DST_INTR_POL, respectively.

For destination interrupts it is possible to drive the output pin permanently or emulate open-collector output.

To drive permanently configure EXT_INTR_DRV[e] = 0.

To emulate open collector output configure EXT_INTR_DRV[e] = 1 and EXT_INTR_POL[e] = 0. To safely enable open-collector output, the EXT_INTR_DRV and EXT_INTR_POL registers must be configured before enabling the overlaid function in the GPIO controller.

Note: Open-collector output mode is needed when multiple interrupt sources are hooked up to the same interrupt wire on the PCB and the wire is pulled high with a resistor. Each interrupt source can then drive the wire low via open-collector output when they want to signal interrupt.

5.0 OPERATIONAL CHARACTERISTICS

5.1 Absolute Maximum Ratings*

Supply Voltage (**VDD**, **VDD_AL**, **VDD_PLL**, **VDD_ALS*x***) (Note 5-1) . . . . . -0.3V to +1.265V

Supply Voltage (**VDD_IO_*x***, **VDD_AH**, **VDD_AHS**) (Note 5-1) . . . . . -0.3V to +3.63V

Input voltage (all inputs) . . . . . -0.3V to +3.63V

Output voltage (all outputs) . . . . . -0.3V to +3.63V

Storage Temperature . . . . . -55°C to +125°C

Lead Temperature Range . . . . . Refer to JEDEC Spec. J-STD-020

Maximum Junction Temperature (T_J) . . . . . +125°C

HBM ESD Performance . . . . . +/-4 kV

Note 5-1 When powering this device from laboratory or system power supplies, it is important that the absolute maximum ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes on their outputs when AC power is switched on or off. In addition, voltage transients on the AC power line may appear on the DC output. If this possibility exists, it is suggested to use a clamp circuit.

*Stresses exceeding those listed in this section could cause permanent damage to the device. This is a stress rating only. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at any condition exceeding those indicated in Section 5.2, "Operating Conditions**", Section 5.5, "DC Specifications", or any other applicable section of this specification is not implied.

5.2 Operating Conditions**

Supply Voltage (**VDD, VDD_AL, VDD_PLL, VDD_ALS*x***) . . . . . +1.15V -5%/+5%

Supply Voltage (**VDD_AH, VDD_AHS**) . . . . . +2.5/3.3V -5%/+5%

Supply Voltage (**VDD_IO_A/B/C/E**) . . . . . +1.8/2.5/3.3V +5%/+5%

Supply Voltage (**VDD_IO_D**) . . . . . +3.3V -5%/+5%

Ambient Operating Temperature in Still Air (T_A) . . . . . T_A= -40°C to T_A= 85°C Industrial

**Proper operation of the device is guaranteed only within the ranges specified in this section. Customers are advised to ensure the maximum junction temperature as specified above is not exceeded when choosing the system power level.

Note: Do not drive input signals without power supplied to the device.

5.3 Package Thermal Specifications

TABLE 5-1: 128-PIN TQFP PACKAGE THERMAL PARAMETERS

Parameter	Symbol	Value	Units	Description
Thermal Resistance Junction to Ambient	Θ_{JA}	15.146	°C/W	**Note 5-2** Measured in still air
		11.47	°C/W	Airflow 1 m/s
		10.42	°C/W	Airflow 2.5 m/s
Thermal Resistance Junction to Bottom of Case	Ψ_{JT}	0.055	°C/W	Measured in still air
Thermal Resistance Junction to Top of Case	Θ_{JC}	2.432	°C/W	—
Thermal Resistance Junction to Board	Θ_{JB}	6.124	°C/W	
Thermal Parameter Junction to Board	Ψ_{JB}	5.6	°C/W	Measured in still air

TABLE 5-2: 156-PIN VQFN-DR PACKAGE THERMAL PARAMETERS

Parameter	Symbol	Value	Units	Description
Thermal Resistance Junction to Ambient	Θ_{JA}	15.356	°C/W	**Note 5-3** Measured in still air
		11.68	°C/W	Airflow 1 m/s
		10.60	°C/W	Airflow 2.5 m/s
Thermal Resistance Junction to Bottom of Case	Ψ_{JT}	0.086	°C/W	Measured in still air
Thermal Resistance Junction to Top of Case	Θ_{JC}	3.025	°C/W	—
Thermal Resistance Junction to Board	Θ_{JB}	4.757	°C/W	
Thermal Parameter Junction to Board	Ψ_{JB}	4.33	°C/W	Measured in still air

Note: Thermal parameters are measured or estimated for devices in a multi-layer 2S2P PCB per JESD51.

5.4 Current and Power Consumption

TABLE 5-3: OPERATING CURRENT

Parameter	Symbol	Typical	Maximum (T_{JMAX}=125°C)	Unit
V_{DD} operating current, 1.15V V_{DD_PLL} operating current, 1.15V	I_{DD} I_{DDPLL}	0.71 (Note 4)	0.935	A
V_{DD_AL} operating current, 1.15V	I_{DD_AL}	400	400	mA
V_{DD_ALSx} operating current, 1.15V	I_{DD_ALSx}	25	25	mA
$V_{DD\_AH+ VDD\_AHS}$ operating current, 2.5V	I_{DD_AH25}	455	475	mA
$V_{DD\_AH+ VDD\_AHS}$ operating current, 3.3V	I_{DD_AH33}	490	505	mA
$V_{DD\_IO\_x}$ operating current, 1.8V	$I_{DD\_IO18A/B/C/E}$	40	40	mA
$V_{DD\_IO\_x}$ operating current, 2.5V	$I_{DD\_IO25A/B/C/E}$	40	45	mA
$V_{DD\_IO\_x}$ operating current, 3.3V	$I_{DD\_IO33A/B/C/E}$	65	75	mA
$V_{DD\_IO\_D}$ operating current, 3.3V	I_{DD_IOD}	5	5	mA

Note 4: Figure includes combined VDD core and PLL currents (i.e. I_{DD} + I_{DDPLL}).

Note: The values above are intended for power supply design only.

TABLE 5-4: POWER CONSUMPTION (VDD_AH/VDD_AHS/VDD_IO_*X* @ 3.3V)

Port Configuration	Typical Estimated Power	Maximum Estimated Power (T_{JMAX}=125°C)	Unit
5x Cu, 2x 2.5G SGMII, 2x RGMII	3.2	3.7	W
5x 1G CuPHY + 1x QSGMII	2.9	3.4	W
4x 1G CuPHY + 1x QSGMII + 1x RGMII	2.5	3.0	W
5x 1G CuPHY + 2x 2.5G SGMII	2.9	3.5	W
5x 1G CuPHY + 1x 2.5G SGMII + 1x RGMII	3.1	3.6	W
5x 1G CuPHY	2.9	3.4	W
4x 1G CuPHY+ 1x 2.5G SGMII	2.6	3.1	W
5x 1G CuPHY + 2x 1G SGMII	2.9	3.4	W
5 x 10M CuPHY	1.2	1.4	W
5 x 100M CuPHY	1.2	1.9	W

TABLE 5-5: POWER CONSUMPTION (VDD_AH/VDD_AHS/VDD_IO_*X* @ 2.5V)

Port Configuration	Typical Estimated Power	Maximum Estimated Power (T_{JMAX}=125°C)	Unit
5x Cu, 2x 2.5G SGMII, 2x RGMII	2.6	3.1	W
5x 1G CuPHY + 1x QSGMII	2.4	2.8	W
4x 1G CuPHY + 1x QSGMII + 1x RGMII	2.1	2.5	W
5x 1G CuPHY + 2x 2.5G SGMII	2.5	3.0	W
5x 1G CuPHY + 1x 2.5G SGMII + 1x RGMII	2.5	3.0	W
5x 1G CuPHY	2.4	2.8	W
4x 1G CuPHY+ 1x 2.5G SGMII	2.2	2.6	W
5x 1G CuPHY + 2x 1G SGMII	2.4	2.8	W
5 x 10M CuPHY	1.0	1.1	W
5 x 100M CuPHY	1.0	1.4	W

5.5 DC Specifications

TABLE 5-6: DC ELECTRICAL CHARACTERISTICS

Parameter	Symbol	Min	Typ	Max	Units	Notes
VIS Type Input Buffer						
Input High Voltage	V_{IH}	0.63*V_{ARIO}		V_{ARIO}	V	
Input Low Voltage	V_{IL}	-0.3		0.39*V_{ARIO}	V	
Input Hysteresis	V_{HYS}	110		220	mV	
Schmitt Falling Trip Point	$V_{T-(1.8)}$	0.67	0.8		V	V_{ARIO}=1.8V
	$V_{T-(2.5)}$		1.1		V	V_{ARIO}=2.5V
	$V_{T-(3.3)}$		1.46	1.68	V	V_{ARIO}=3.3V
Schmitt Rising Trip Point	$V_{T+(1.8)}$	0.8	0.94		V	V_{ARIO}=1.8V
	$V_{T+(2.5)}$		1.25		V	V_{ARIO}=2.5V
	$V_{T+(3.3)}$		1.62	1.85	V	V_{ARIO}=3.3V
Input High Current	I_H			10	µA	V_I=V_{ARIO} (Note 5-4)
Input Low Current	I_L	-10			µA	V_I=0V (Note 5-4)
Internal Pull-Up Resistor	R_{DPU}	58.9	71.5/71/70.7	83.7	kΩ	
Internal Pull-Down Resistor	R_{DPD}	58.7	70.9/71.1/71.2	84.5	kΩ	
VO Type Output Buffer						
Output High Voltage	V_{OH}	V_{ARIO}-0.4			V	I_{OH}=2/4/8/10 mA
Output Low Voltage	V_{OL}			0.4	V	I_{OL}=2/4/8/10 mA
I/O Sinking DC Current	I_{OL2}	1.9		5.0	mA	V_{OUT}=0.4V Drive=2 mA
	I_{OL4}	3.9		10	mA	V_{OUT}=0.4V Drive=4 mA
	I_{OL8}	7.9		20	mA	V_{OUT}=0.4V Drive=8 mA
	I_{OL10}	9.9		25	mA	V_{OUT}=0.4V Drive=10 mA
I/O Sourcing DC Current	I_{OH2}	1.9		4.7	mA	V_{OUT}=V_{ARIO} -0.4V, Drive=2 mA
	I_{OH4}	3.9		9.4	mA	V_{OUT}=V_{ARIO} -0.4V, Drive=4 mA
	I_{OH8}	8.0		18	mA	V_{OUT}=V_{ARIO} -0.4V, Drive=8 mA
	I_{OH10}	9.9		22	mA	V_{OUT}=V_{ARIO} -0.4V, Drive=10 mA
IS Type Input Buffer						
Input High Voltage	V_{IH}	0.63*$V_{DD\_IO\_D}$		$V_{DD\_IO\_D}$	V	
Input Low Voltage	V_{IL}	-0.3		0.39*$V_{DD\_IO\_D}$	V	
Input Hysteresis	V_{HYS}	110		220	mV	
Schmitt Falling Trip Point	$V_{T-(3.3)}$		1.46	1.68	V	$V_{DD\_IO\_D}$=3.3V
Schmitt Rising Trip Point	$V_{T+(3.3)}$		1.62	1.85	V	$V_{DD\_IO\_D}$=3.3V
Input High Current	I_H			10	µA	V_I=$V_{DD\_IO\_D}$ (Note 5-4)
Input Low Current	I_L	-10			µA	V_I=0V (Note 5-4)

TABLE 5-6: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	Min	Typ	Max	Units	Notes
Internal Pull-Up Resistor	R_{DPU}	58.9	71.5/71/70.7	83.7	kΩ	
Internal Pull-Down Resistor	R_{DPD}	58.7	70.9/71.1/71.2	84.5	kΩ	
O Type Output Buffer						
Output High Voltage	V_{OH}	$V_{DD\_IO\_D}$-0.4			V	I_{OH}=2 /4/8/10 mA
Output Low Voltage	V_{OL}			0.4	V	I_{OL}=2/4/8/10 mA
I/O Sinking DC Current	I_{OL2}	1.9		5.0	mA	V_{OUT}=0.4V Drive=2 mA
	I_{OL4}	3.9		10	mA	V_{OUT}=0.4V Drive=4 mA
	I_{OL8}	7.9		20	mA	V_{OUT}=0.4V Drive=8 mA
	I_{OL10}	9.9		25	mA	V_{OUT}=0.4V Drive=10 mA
I/O Sourcing DC Current	I_{OH2}	1.9		4.7	mA	V_{OUT}=$V_{DD\_IO\_D}$ -0.4V, Drive=2 mA
	I_{OH4}	3.9		9.4	mA	V_{OUT}=$V_{DD\_IO\_D}$ -0.4V, Drive=2 mA
	I_{OH8}	8.0		18	mA	V_{OUT}=$V_{DD\_IO\_D}$ -0.4V, Drive=2 mA
	I_{OH10}	9.9		22	mA	V_{OUT}=$V_{DD\_IO\_D}$ -0.4V, Drive=2 mA
HS_I Type Input Buffer						
Input High Voltage	V_{IH}	0.6*V_{ARIO}		V_{ARIO}	V	RGMII mode
Input Low Voltage	V_{IL}	-0.3		0.4*V_{ARIO}	V	RGMII mode
Input Hysteresis	V_{HYS}	60		150	mV	
Schmitt Falling Trip Point	$V_{T-(1.8)}$	0.76	0.9		V	V_{ARIO}=1.8V
	$V_{T-(2.5)}$		1.11		V	V_{ARIO}=2.5V
	$V_{T-(3.3)}$		1.42	1.64	V	V_{ARIO}=3.3V
Schmitt Rising Trip Point	$V_{T+(1.8)}$	0.85	0.99		V	V_{ARIO}=1.8V
	$V_{T+(2.5)}$		1.23		V	V_{ARIO}=2.5V
	$V_{T+(3.3)}$		1.55	1.76	V	V_{ARIO}=3.3V
Input High Current	I_H			10	µA	V_I=V_{ARIO} (Note 5-4)
Input Low Current	I_L	-10			µA	V_I=0V (Note 5-4)
Internal Pull-Up Resistor	R_{DPU}	58.9	71.5/71/70.7	83.7	kΩ	
Internal Pull-Down Resistor	R_{DPD}	58.7	70.9/71.1/71.2	84.5	kΩ	
HS_O Type Output Buffer						
Output High Voltage	V_{OH}	V_{ARIO}-0.4			V	I_{OH}=5 mA
Output Low Voltage	V_{OL}			0.4	V	I_{OL}=5 mA
Output Sink DC Current	I_{OL}	5			mA	V_{OUT}=0.4V
Output Source DC Current	I_{OH2}	5			mA	V_{OUT}=V_{ARIO}-0.4V
Output Source Impedance	R_O		50		Ω	

Note 5-4 Input high current and input low current equals the maximum leakage current, excluding the current in the built-in pull resistors.

Note 5-5 Typical values for V_{ARIO}=1.8/2.5/3.3V, respectively.

Note 5-6 Differential output swing is register configurable.

5.5.1 SERDES

TABLE 5-7: SERDES RECEIVER (SGMII, SFP, 1000BASE-KX, 2.5G, 2.5GBASE-KX, QSGMII)

Parameter	Symbol	Min	Typ	Max	Units	Notes
5G_I Type Input Buffer						
Differential input resistance	R_{DIFF}	80	100	120	Ω	SGMII, SGMII2G5, QSGMII
Input voltage range	V_{IP}, V_{IN}	0		VDD_ALS*x* + 300	mV	SGMII, SGMII2G5, QSGMII
Differential peak-to-peak input swing	$\|V_{ID}\|$	120		1200	mVppd	SGMII, SGMII2G5, QSGMII
Input common-mode voltage	V_{CM}	$\|V_{ID}\|$ / 2		VDD_ALS*x*	mV	

TABLE 5-8: SERDES TRANSMITTER (SGMII, SFP, 1000BASE-KX, 2.5G, 2.5GBASE-KX, QSGMII)

Parameter	Symbol	Min	Typ	Max	Units	Notes
5G_O Type Output Buffer						
Differential resistance	R_{DIFF}	80	100	120	Ω	
Differential peak-to-peak output voltage (Note 5-7)	V_{O_DIFF}	300		800	mVppd	SGMII, SFP, 2.5G, QSGMII
Differential peak-to-peak output voltage (Note 5-7)	V_{O_DIFF}	800		1300	mVppd	1000BASE-KX, 2.5GBASE-KX
DC Common mode voltage	$V_{TX\_DC\_CM}$	0		1	V	

Note 5-7 Differential output swing is register configurable

TABLE 5-9: SERDES REFERENCE CLOCK INPUT (S0_CLKP)

Parameter	Symbol	Min	Typ	Max	Units
Input high voltage	V_{IH}	VDD_ALS0 - 300		VDD_ALS0 + 300	mV
Input low voltage	V_{IL}	-300		300	mV

5.6 AC Specifications

This section details the various AC timing specifications of the device.

5.6.1 REFERENCE CLOCKS

5.6.1.1 XI Reference Clock Timing Specifications

TABLE 5-10: XI TIMING SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
REFCLK Frequency REFCLK_SEL = 100	f	-100 ppm	25	+2000 ppm	MHz	
Clock duty cycle		40		60	%	Measured at 50% threshold
Rise time and fall time	t_R, t_F			2.2	ns	10%-90% 3.3V signal
REFCLK input peak-to-peak jitter, bandwidth from 2.5 kHz to 10 MHz (Note 5-8)				20	ps	

Note 5-8 Peak-to-peak values are typically higher than the RMS value by a factor of 10 to 14.

5.6.1.2 SerDes Reference Clock (S0_CLKP) Timing Specifications

TABLE 5-11: SERDES REFERENCE CLOCK (S0_CLKP) TIMING SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
REFCLK Frequency	f	-100 ppm	125	+100 ppm	MHz	
Clock duty cycle		40		60	%	Measured at 50% threshold
Input edge rate		0.4		2	V/ns	Differential +/-250 mV threshold
Cycle-to-cycle jitter				150	ps	D_J across all frequencies
Random jitter	R_J			3	ps (RMS)	1.5 MHz to Nyquist frequency (62.5 MHz).
				1.6	ps (RMS)	Integrated R_J from 12 kHz to 20 MHz
				1.2	ps (RMS)	Integrated R_J from 2 MHz to 20 MHz
Deterministic jitter	D_J			5.6	ps (peak-peak)	200 kHz to 50 MHz
				2.8	ps (peak-peak)	750 kHz to 10 MHz
Jitter transfer from REFCLK to SerDes output, bandwidth from 10 kHz to 1 MHz				0.3	ps	

TABLE 5-11: SERDES REFERENCE CLOCK (S0_CLKP) TIMING SPECIFICATIONS (CONTINUED)

Parameter	Symbol	Min	Typ	Max	Units	Notes
Jitter transfer from REFCLK to SerDes output, bandwidth from 1 MHz to 10 MHz				2	dB	
Jitter transfer from REFCLK to SerDes output, bandwidth above 10 MHz				2 - 20 x log (f/10 MHz)	dB	
REFCLK input peak-to-peak jitter, bandwidth from 2.5 kHz and 10 MHz (Note 5-9)				20	ps	To meet G.8262 1G SyncE jitter generation spec.

Note 5-9 Peak-to-peak values are typically higher than the RMS value by a factor of 10 to 14.

5.6.2 SERDES

5.6.2.1 Transmitter (SGMII, SFP, 1000BASE-KX, 2.5G, 2.5BASE-KX) Timing Specifications

TABLE 5-12: SERDES TRANSMITTER TIMING SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
Data rate		1.25 - 100 ppm	1.25	1.25 + 100 ppm	Gbps	SGMII (Cisco), SFP (SFP-MSA), 1000BASE-KX (IEEE 802.3, clause 70)
Data rate		3.125 - 100 ppm	3.125	3.125 + 100 ppm	Gbps	2.5G, 2.5GBASE-KX (IEEE 802.3, clause 128)
Differential output return loss	RLO_{SDD22}			-10 -10 + 10 x log(f/625 MHz)	dB	1000BASE-KX: 50 MHz to 625 MHz 625 MHz to 1250 MHz
Differential output return loss	RLO_{SDD22}			-10 -10 + 10 x log(f/625 MHz)	dB	2.5GBASE-KX: 100 MHz to 625 MHz 625 MHz to 2000MHz
Common-mode output return loss	RLO_{SCC22}			-7 -7 + 10 x log(f/625 MHz)	dB	2.5GBASE-KX: 100 MHz to 625 MHz 625 MHz to 2000 MHz
Rise and fall time	t_R, t_F	60		320	ps	20% to 80%, 1000BASE-KX
Rise and fall time	t_R, t_F	30		100	ps	20% to 80%, 2.5GBASE-KX
Interpair skew	t_{SKEW}			20	ps	
Random jitter	R_J			0.15	UI_{P-P}	At BER 10^{-12}, 1000BASE-KX
Random jitter	R_J			0.2	UI_{P-P}	2.5GBASE-KX
Deterministic jitter	D_J			0.10	UI_{P-P}	1000BASE-KX
Deterministic jitter	D_J			0.12	UI_{P-P}	Duty cycle distortion of 0.035 UI is considered part of D_J, 2.5GBASE-KX, 5G-Q(U)SGMII

TABLE 5-12: SERDES TRANSMITTER TIMING SPECIFICATIONS (CONTINUED)

Parameter	Symbol	Min	Typ	Max	Units	Notes
Total jitter	T_J			0.25	UI_{P-P}	At BER 10^{-12}, 1000BASE-KX
Total jitter	T_J			0.32	UI_{P-P}	At BER 10^{-12}, 2.5GBASE-KX
Wideband SyncE jitter	T_{WJ}			0.5	UI_{P-P}	Measured according to ITU-T G.8262, section 8.3
Eye mask	X1			0.125	UI	
Eye mask	X2			0.325	UI	
Eye mask	Y1	350			mV	
Eye mask	Y2			800	mV	

5.6.2.2 Receiver (SGMII, SFP, 1000BASE-KX, 2.5G, 2.5GBASE-KX) Timing Specifications

TABLE 5-13: SERDES RECEIVER TIMING SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
Data rate		1.25 - 100 ppm	1.25	1.25 + 100 ppm	Gbps	SGMII, SFP, 1000BASE-KX.
Data rate		3.125 – 100 ppm	3.125	3.125 + 100 ppm	Gbps	2.5GBASE-KX.
Differential input return loss	RLI_{SDD11}			-10	dB	50 MHz to 625 MHz
Differential input return loss	RLI_{SDD11}			-10 + 10 x log(f/625 MHz)	dB	625 MHz to 2000 MHz
Common-mode input return loss	RLO_{SCC11}			-7	dB	100 MHz to 625 MHz
Common-mode input return loss	RLO_{SCC11}			-7 + 10 x log(f/ 625 MHz)	dB	625 MHz to 2000 MHz
Jitter tolerance, total (Note 5-10)	TOL_{TJ}	600			ps	Measured according to IEEE 802.3 Clause 38.6.8
Jitter tolerance, deterministic (Note 5-10)	TOL_{DJ}	370			ps	Measured according to IEEE 802.3 Clause 38.6.8
Wideband SyncE jitter tolerance	WJT	312.5			UI_{P-P}	10 Hz to 12.1 Hz. Measured according to ITU-T G.8262, section 9.2.
Wideband SyncE jitter tolerance	WJT	3750/f			UI_{P-P}	12.1 Hz to 2.5 kHz (f). Measured according to ITU-T G.8262, section 9.2.

TABLE 5-13: SERDES RECEIVER TIMING SPECIFICATIONS (CONTINUED)

Parameter	Symbol	Min	Typ	Max	Units	Notes
Wideband SyncE jitter tolerance	WJT	1.5			UI_{P-P}	2.5 kHz to 50 kHz. Measured according to ITU-T G.8262, section 9.2.

Note 5-10 Jitter requirements represent high-frequency jitter (above 637kHz) and not low-frequency jitter or wander.

5.6.2.3 QSGMII Transmitter

TABLE 5-14: QSGMII TRANSMITTER SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
Data rate		5.0 – 100ppm	5.0	5.0 + 100 ppm	Gbps	
Differential output return loss	RLO_{SDD22}			-8	dB	100 MHz to 2.5 GHz
Differential output return loss	RLO_{SDD22}			-8 + 16.6 x log(f/2.5 GHz)	dB	2.5 GHz to 5.0 GHz
Common-mode output return loss	RLO_{SCC22}			-6	dB	100 MHz to 2.5 GHz
Rise time and fall time	t_R, t_F	30		130	ps	20% to 80%. Recommended value.
Random jitter	RJ			0.15	UI_{P-P}	
Deterministic jitter	DJ			0.15	UI_{P-P}	
Duty cycle distortion (part of DJ)	DCD			0.05	UI_{P-P}	
Total jitter	TJ			0.3	UI_{P-P}	
Eye mask	X1			0.15	UI_{P-P}	Near-end
Eye mask	X2			0.4	UI_{P-P}	Near-end
Eye mask	Y1	200			mV	Near-end
Eye mask	Y2			450	mV	Near-end

Note 5-11 DC coupling is not supported by the SerDes in QSGMII mode.

5.6.2.4 QSGMII Receiver

TABLE 5-15: QSGMII RECEIVER SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
Data rate		5.0 – 100 ppm	5.0	5.0 + 100 ppm	Gbps	
Differential input return loss	RLI_{SDD11}			-8	dB	100 MHz to 2.5 GHz
Differential input return loss	RLI_{SDD11}			-8 + 16.6 x log(f/2.5 GHz)	dB	2.5 GHz to 5.0 GHz

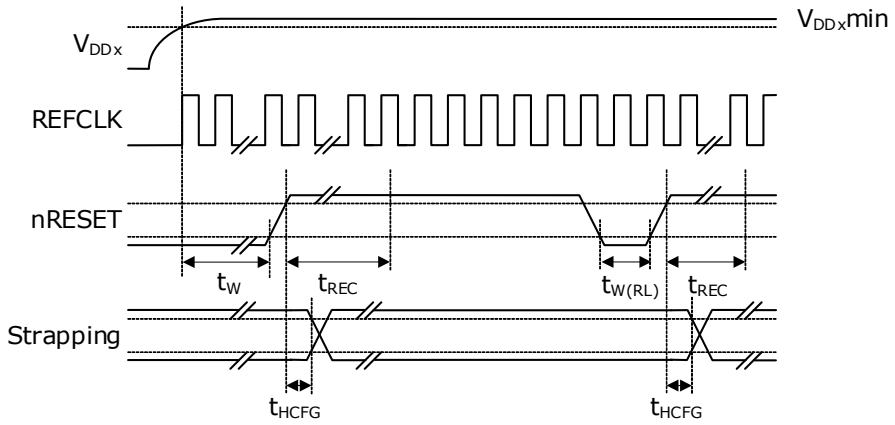
TABLE 5-15: QSGMII RECEIVER SPECIFICATIONS (CONTINUED)

Parameter	Symbol	Min	Typ	Max	Units	Notes
Common-mode return loss	RLI_{SCC11}			-6	dB	100 MHz to 2.5 GHz
Sinusoidal jitter maximum	SJ_{MAX}			5	UI_{P-P}	For low sinusoidal jitter frequencies below (baud/1667)
Sinusoidal jitter, high frequency	SJ_{HF}			0.05	UI_{P-P}	
Deterministic jitter (uncorrelated bounded high-probability jitter)	UBHPJ			0.15	UI_{P-P}	
Data-dependent jitter (correlated bounded high-probability jitter)	CBHPJ			0.3	UI_{P-P}	
Total jitter	TJ			0.6	UI_{P-P}	Sinusoidal jitter excluded
Eye mask	R_X1			0.3	UI_{P-P}	
Eye mask	R_Y1	50			mV	
Eye mask	R_Y2			450	mV	

5.6.3 NRESET TIMING SPECIFICATIONS

The nRESET signal waveform and the required measurement points for the timing specification are shown in Figure 5-1.

FIGURE 5-1: NRESET TIMING DIAGRAM



The signal applied to the nRESET input must comply with the specifications listed in the following table at the reset pin of the devices.

TABLE 5-16: NRESET TIMING SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
nRESET assertion time after power supplies and clock stabilizes	t_W	2			ms	
nRESET pulse width	$t_{W(RL)}$	100			ns	
Hold time for GPIO-mapped strapping pins relative to nRESET	t_{HCFG}	50			ns	
Recovery time from reset inactive to device active (starts reading configuration PROM)	t_{REC}			100	ms	
Recovery time from reset inactive to device fully active (SYSTEM READY signal active in unmanaged device)			500		ns	

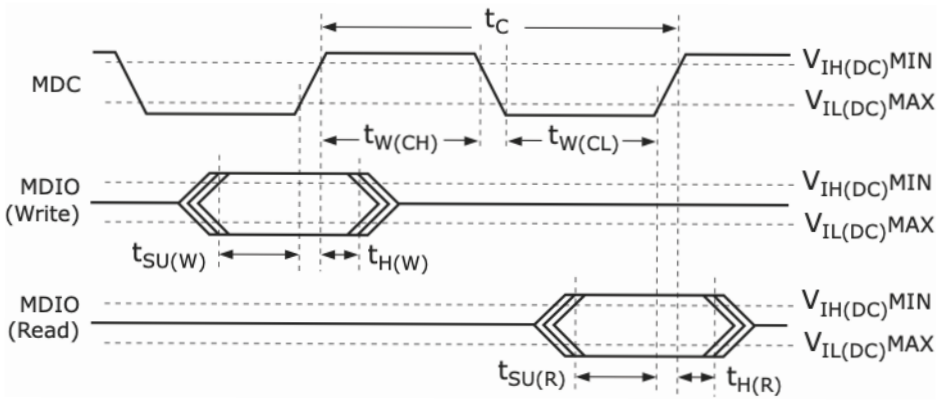
5.6.4 MIIM TIMING SPECIFICATIONS

All AC specifications for the MII Management (MIIM) interface meet or exceed the requirements of IEEE 802.3-2002 (clause 22.2-4).

All MIIM AC timing requirements are specified relative to the input low and input high threshold levels.

The following illustration shows the MIIM waveforms and required measurement points for the signals.

FIGURE 5-2: MIIM TIMING DIAGRAM



The set up time of MDIO relative to the rising edge of MDC is defined as the length of time between when the MDIO exits and remains out of the switching region and when MDC enters the switching region. The hold time of MDIO relative to the rising edge of MDC is defined as the length of time between when MDC exits the switching region and when MDIO enters the switching region.

All MIIM signals comply with the specifications in the following table. The MDIO signal requirements are requested at the pin of the devices.

TABLE 5-17: MIIM TIMING SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
MDC frequency	f	0.323 (Note 5-14)	1.624	41.4	MHz	(Note 5-12)
MDC cycle time	t_C	24		3092 (Note 5-14)	ns	(Note 5-13)
MDC time high	$t_{W(CH)}$	10			ns	C_L=50 pF
MDC time low	$t_{W(CL)}$	10			ns	C_L=50 pF
MDC setup time to MDC on write	$t_{SU(W)}$	$t_{W(CL)}^{-2}$			ns	C_L=50 pF
MDIO hold time from MDC on write	$t_{H(W)}$	$t_{W(CH)}^{-2}$			ns	C_L=50 pF
MDIO setup time to MDC on read	$t_{SU(R)}$	14			ns	C_L=50 pF on MDC
MDIO hold time from MDC on read	$t_{H(R)}$	0			ns	C_L=50 pF

Note 5-12 For the maximum value, the device supports an MDC clock speed of up to 20 MHz for faster communication with the PHYs. If the standard frequency of 2.5 MHz is used, the MIIM interface is designed to meet or exceed the IEEE 802.3 requirements of the minimum MDC high and low times of 160 ns and an MDC cycle time of minimum 400 ns, which is not possible at faster speeds.

Note 5-13 Calculated as $f_C = 1/f$.

Note 5-14 Value is for 165.625 MHz core clock.

FIGURE 5-3: MIIM CLIENT TIMING DIAGRAM

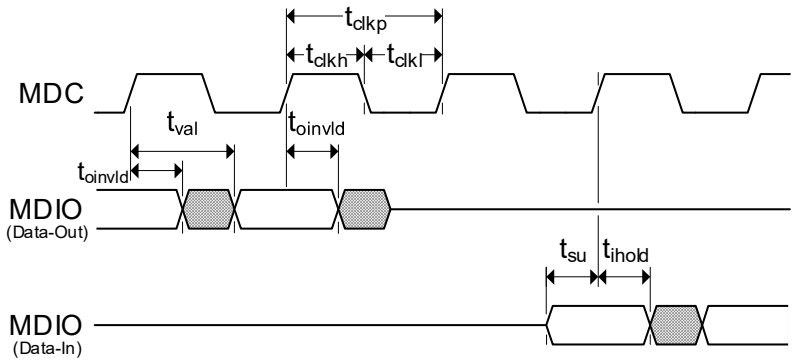


TABLE 5-18: MIIM CLIENT TIMING SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
MDC frequency	f		2.5	20	MHz	Speed is limited by MDIO external pull-up
MDC cycle time	t_{CLKP}	50			ns	
MDC time high	t_{CLKH}	20			ns	C_L=50 pF
MDC time low	t_{CLKL}	20			ns	C_L=50 pF
MDC setup time to MDC on write	t_{SU}	14			ns	C_L=50 pF
MDIO hold time from MDC on write	t_{IHOLD}	8			ns	C_L=50 pF
MDIO output valid on read	t_{VAL}			42	ns	C_L=50 pF on MDC
MDIO output hold time on read	t_{OINVLD}	28			ns	C_L=50 pF

5.6.5 RGMII TIMING SPECIFICATIONS

TABLE 5-19: RGMII TIMING SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
RXC/TXC clock frequency (1000 Mbps)	f		125		MHz	
Clock duty cycle (1000 Mbps)	Duty_G	45		55	%	
Clock cycle duration (1000 Mbps)	T_{CYC}	7.2	8	8.8	ns	
RXC/TXC clock frequency (100 Mbps)			25		MHz	
Clock duty cycles (10/100 Mbps)	Duty_T	40		60	%	
RGMII signal rise/fall time (2.5V/3.3V)	T_r/T_f			0.75	ns	20%-80%, 2.5V/3.3V
RGMII signal rise/fall time (1.8V)	T_r/T_f			1.0	ns	20%-80%, 1.8V
Data to clock input setup time	t_{SETUP}	2.2			ns	(Note 5-15)
Data to clock input hold time	t_{HOLD}	0			ns	(Note 5-15)
Data to clock output skew	t_{SKEW}	1.1	2.0		ns	(Note 5-16)

Note 5-15 For cases where there is no (or insufficient) skew between the input data and input clock, it is possible to add internal delay to the TXC signal by setting a configuration bit. This feature reduces the setup time requirement and increases the hold time requirement nominally by 1.3 ns.

Note 5-16 The RGMII interface adheres to the RGMII Specification Version 2.0, which specifies the driving device delay the output clock relative to the output data. This is the TSKEW parameter. This skew can be disabled by clearing a configuration bit. Generally this is not recommended.

5.6.6 SERIAL INTERFACE TIMING SPECIFICATIONS

5.6.6.1 SI Timing Specifications (Host Mode)

FIGURE 5-4: SI TIMING DIAGRAM

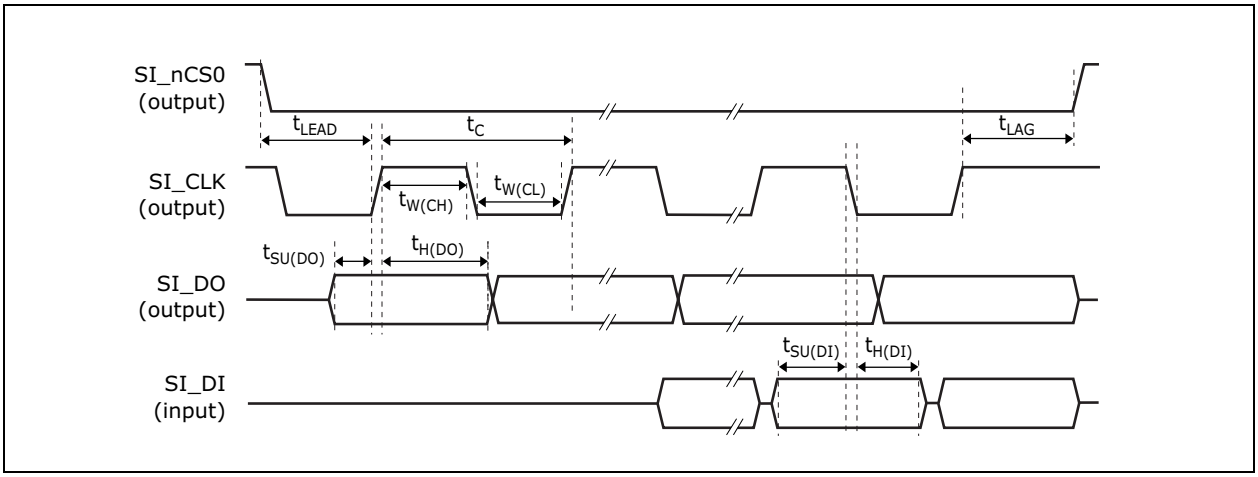


TABLE 5-20: SI TIMING SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
Clock frequency	f			25 (Note 5-17)	MHz	
Clock cycle time	t_C	40			ns	
Clock time high	$t_{W(CH)}$	16			ns	
Clock time low	$t_{W(CL)}$	16			ns	
Clock rise time and fall time	t_R,T_F	1		5	ns	Between $V_{IL(MAX)}$ and $VI_{H(MIN)}$. C_L=5 pF.
SI_DO output valid time from previous clock transition	$t_{SU(DO)}$	$t_{w\ (CL)}{}^{-2}$			ns	
SI_DO output hold time from next clock transition	$t_{H(DO)}$	$t_{w\ (CH)}{}^{-2}$			ns	
Enable active before first clock	t_{LEAD}	$t_c{}^{/2+22}$			ns	
Enable inactive after clock	t_{LAG}	$t_c{}^{/2+17}$			ns	
SI_DI input setup time to clock falling edge	$t_{SU(DI)}$	11			ns	
SI_DI input hold time from clock falling edge	$t_{H(DI)}$	6			ns	

Note 5-17 Frequency is programmable.

5.6.6.2 SI Timing Specifications (Client Mode)

FIGURE 5-5: SI INPUT DATA TIMING DIAGRAM FOR CLIENT MODE

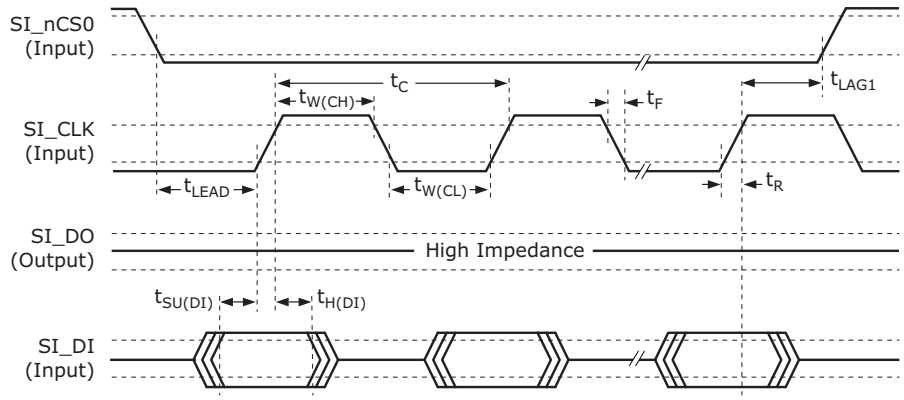


FIGURE 5-6: SI OUTPUT DATA TIMING DIAGRAM FOR CLIENT MODE

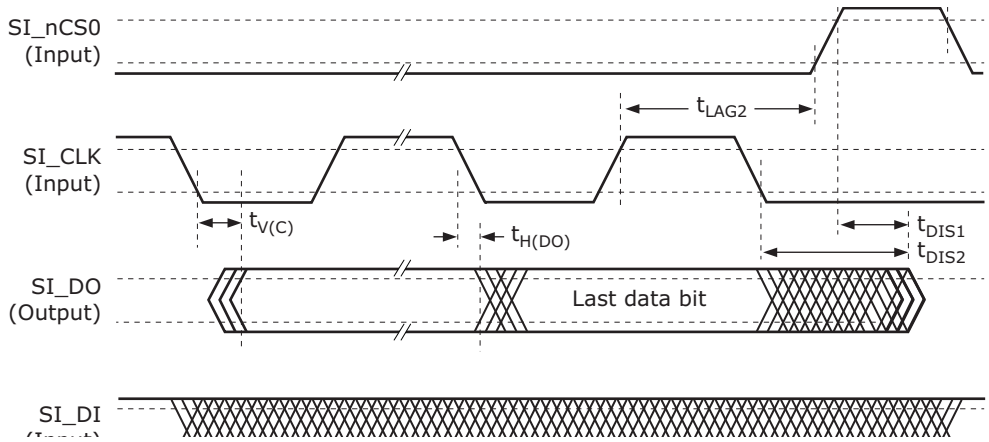


TABLE 5-21: SI CLIENT TIMING SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
Clock frequency	f			25	MHz	
Clock cycle time	t_C	40			ns	
Clock time high	$t_{W(CH)}$	16			ns	
Clock time low	$t_{W(CL)}$	16			ns	
SI_DI input setup time to clock rising edge	$t_{SU(DI)}$	2			ns	
SI_DI input hold time from clock rising edge	$t_{H(DI)}$	2			ns	
Enable active before first clock	t_{LEAD}	10			ns	
Enable inactive after clock (input cycle)	t_{LAG1}	16			ns	(Note 5-18)
Enable inactive after clock (output cycle)	t_{LAG2}	(Note 5-19)			ns	
Enable inactive width	$t_{W(EH)}$	40			ns	

TABLE 5-21: SI CLIENT TIMING SPECIFICATIONS (CONTINUED)

Parameter	Symbol	Min	Typ	Max	Units	Notes
SI_DO output valid after clock falling edge	$t_{V(C)}$			12	ns	C_L=30 pF
SI_DO output hold time from clock falling edge	$t_{H(DO)}$	10			ns	C_L=30 pF
SI_DO disable time	t_{DIS1}			12	ns	(Note 5-20)
SI_DO disable time	t_{DIS2}			12	ns	(Note 5-20)

Note 5-18 t_{LAG1} is defined only for write operations to the devices, not for read operations.

Note 5-19 The last rising edge on the clock is necessary for the external master to read in the data. The lag time depends on the necessary hold time on the external master data input.

Note 5-20 Pin begins to float when a 300 mV change from the loaded V_{OH} or V_{OL} level occurs.

5.6.7 JTAG INTERFACE

FIGURE 5-7: JTAG TIMING DIAGRAM

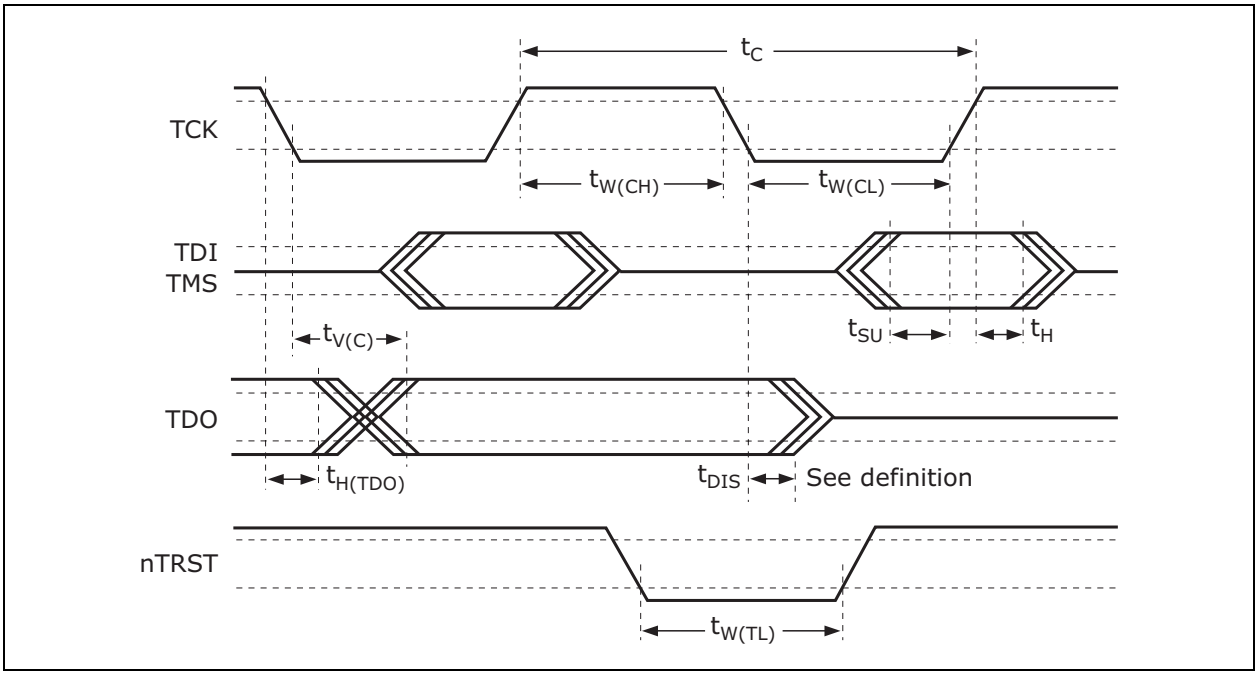


TABLE 5-22: JTAG TIMING SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
TCK frequency	f			10	MHz	
TCK cycle time	t_C	100			ns	
TCK high time	$t_{W(CH)}$	40			ns	
TCK low time	$t_{W(CL)}$	40			ns	
TDI setup time to TCK rising	t_{SU}	5			ns	
TDI hold time from TCK rising	t_H	5			ns	
TDO valid after TCK falling	$t_{V(C)}$			16	ns	C_L=10 pF
TDO hold after from TCK falling	$t_{H(TDO)}$	12			ns	C_L=0 pF
TDO disable time	t_{DIS}			16	ns	(Note 5-21)
nTRST time low	$t_{W(TL)}$	30			ns	

Note 5-21 The pin begins to float when a 300 mV change from the actual V_{OH}/V_{OL} level occurs.

5.6.8 SERIAL I/O TIMING SPECIFICATIONS

FIGURE 5-8: SERIAL I/O TIMING

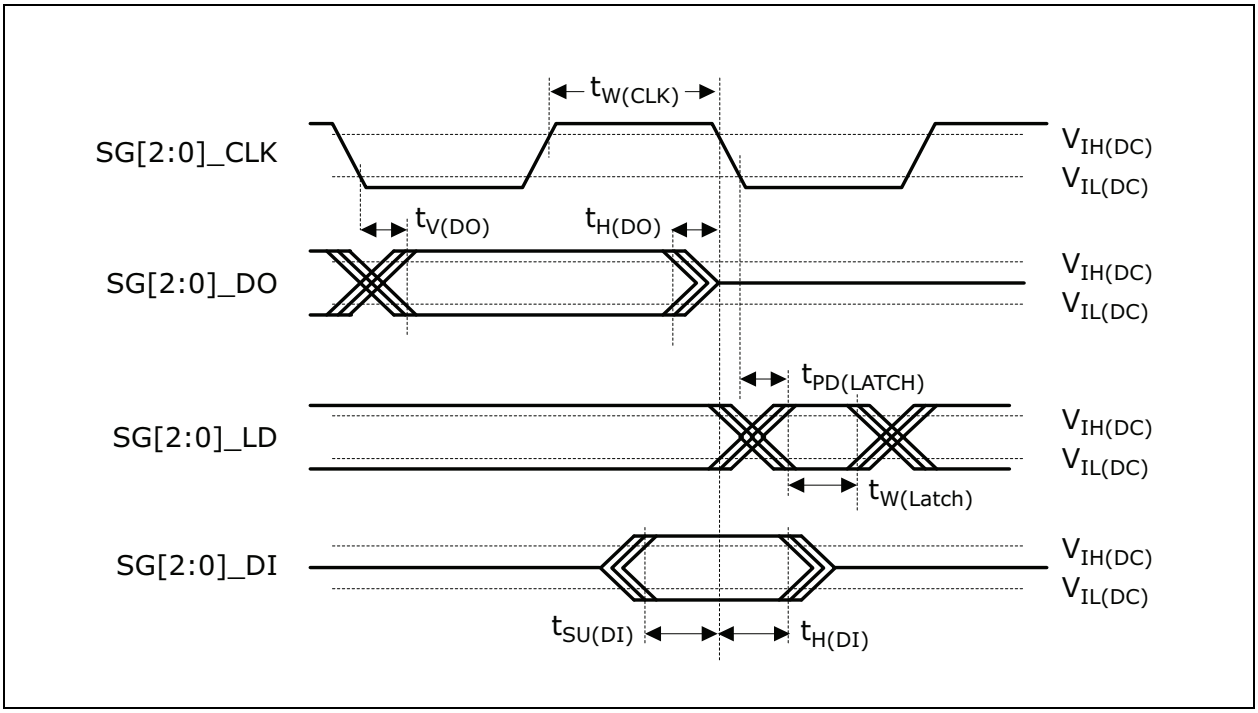


TABLE 5-23: SERIAL I/O TIMING SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
Clock frequency	f			25	MHz	(Note 5-22)
SG0_CLK clock pulse width	$t_{W(CLK)}$	40		3	%	
SG0_DO valid after clock falling	$t_{V(DO)}$			8	ns	
SG0_DO hold time from clock falling	$t_{H(DO)}$			-2	ns	
SG0_LD propagation delay from clock falling	$t_{PD(LATCH)}$	40			ns	
SG0_LD width	$t_{W(LATCH)}$	10			ns	
SG0_DI setup time to clock falling	$t_{SU(DI)}$	13			ns	
SG0_DI hold time from clock falling	$t_{H(DI)}$	-7			ns	

Note 5-22 The SIO clock frequency is programmable.

5.6.9 RECOVERED CLOCK OUTPUT TIMING SPECIFICATIONS

This section provides the AC characteristics for the recovered clock output signals.

Figure 5-9 shows the test circuit for the recovered clock output signals.

FIGURE 5-9: TEST CIRCUIT FOR RECOVERED CLOCK OUTPUT SIGNALS

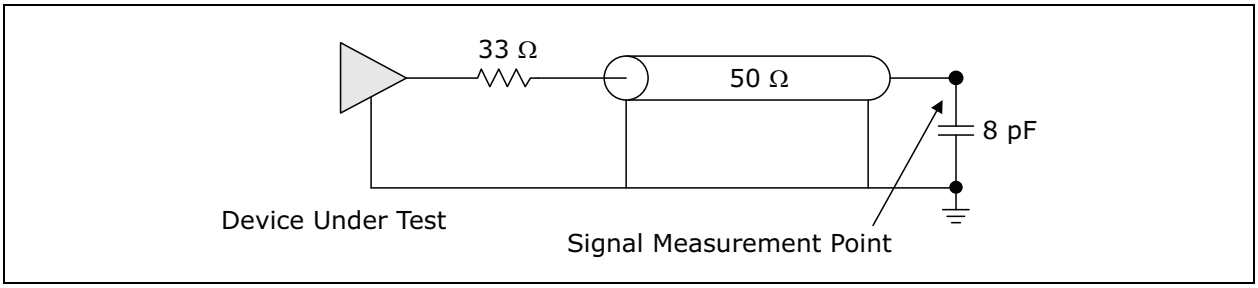


Table 5-24 lists the AC specifications for the recovered clock outputs.

TABLE 5-24: RECOVERED CLOCK OUTPUT TIMING SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
RCLK[1:0] clock frequency (on GPIO[9:10])	f			125	MHz	
RCLK[1:0] clock frequency (on GPIO[28:29], GPIO[35:34])	f			60	MHz	
Clock duty cycles	t_C	40		60	%	Measured at 50% threshold
RCLK[1:0] rise/fall time	t_R, t_F			2	ns	
Squelching delay from SerDes signal to RCLK[1:0]				200	ns	Squelch enabled
RCLK[1:0] peak-to-peak jitter, bandwidth between 12 KHz and 10 MHz, 60 second gate time				200	ps	Jitter-free input to SerDesRX (Note 5-23)
RCLK[1:0] peak-to-peak jitter, bandwidth between 10 MHz and 80 MHz, 60 second gate time				200	ps	Jitter-free input to SerDes RX (Note 5-23)

Note 5-23 Maximum jitter on the recovered signal.

5.6.10 TWO-WIRE SERIAL INTERFACE TIMING SPECIFICATIONS

FIGURE 5-10: TWO-WIRE SERIAL READ TIMING DIAGRAM

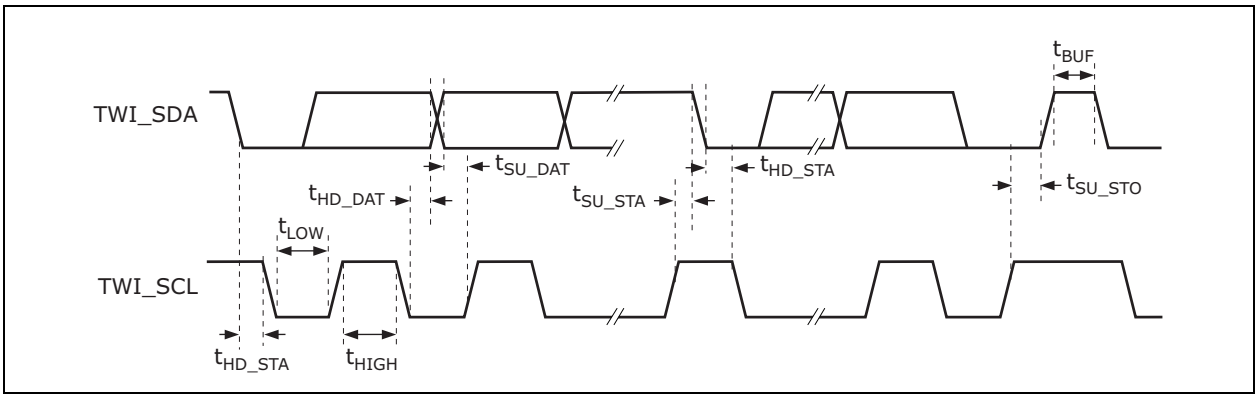


FIGURE 5-11: TWO-WIRE SERIAL WRITE TIMING DIAGRAM

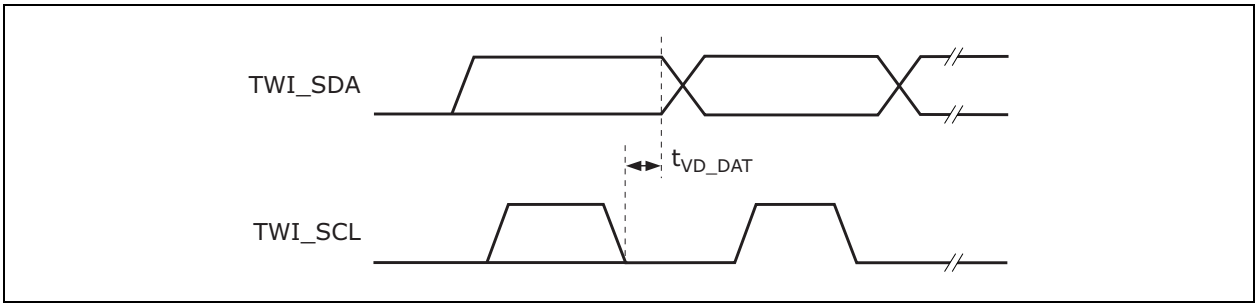


TABLE 5-25: TWO-WIRE SERIAL TIMING SPECIFICATIONS

Parameter	Symbol	Standard Mode		Fast Mode		Fast Mode+		Units
		Min	Max	Min	Max	Min	Max	
TWI_SCL clock frequency	f	0	100	0	400	0	1000	kHz
TWI_SCL low period	t_{LOW}	4.7		1.3		0.5		µs
TWI_SCL high period	t_{HIGH}	4.0		0.6		0.26		µs
TWI_SCL and TWI_SDA rise time			1000	20	300		120	ns
TWI_SCL and TWI_SDA fall time			300		300		120	ns
TWI_SDA setup time to TWI_SCL fall	t_{SU_DAT}	250		100		50		ns
TWI_SDA hold time to TWI_SCL fall (Note 5-24)	t_{HD_DAT}	300	3450	300	900		450	ns
Setup time for repeated START condition	t_{SU_STA}	4.7		0.6		0.26		µs
Hold time after repeated START condition	t_{HD_STA}	4.0		0.6		0.26		µs
Bus free time between STOP and START conditions	t_{BUF}	4.7		1.3		0.5		µs
Clock to valid data out (Note 5-25)	t_{VD_DAT}		3450		900		450	ns
Clock to valid data acknowledge	t_{VD_ACK}		3450		900		450	ns

Note 5-24 An external device must provide a hold time of at least 300 ns for the TWI_SDA signal to bridge the undefined region of the falling edge of the TWI_SCL signal.

Note 5-25 Some external devices may require more data in hold time (target device's t_{HD_DAT}) than what is provided by t_{VD_DAT}, for example, 300 ns to 900 ns. The minimum value of t_{VD_DAT} is adjustable; the value given represents the recommended minimum value, which is enabled ICOU_CFG::TWI_CONFIG.TWI_DELAY_ENABLE.

5.6.11 IEEE 1588 TIME TICK OUTPUT TIMING SPECIFICATIONS

TABLE 5-26: RECOVERED CLOCK OUTPUT TIMING SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units	Notes
PTP[3:0] frequency (on GPIO[15:12])	f			125	MHz	(Note 5-26)
PTP[3:0] frequency (on GPIO[33:30,37:36])	f			60	MHz	(Note 5-26)
Clock duty cycle		40		60	%	Measured at 50% threshold (Note 5-27)
PTP[3:0] rise/fall time (on GPIO[15:12])	t_R, t_F			1.5	ns	20% to 80% threshold. 5 pF load.
PTP[3:0] rise/fall time (on GPIO[33:30,37:36])	t_R, t_F			5	ns	20% to 80% threshold. 5 pF load.
PTP[3:0] peak-to-peak jitter, bandwidth between 12 KHz and 10 MHz				200	ps	(Note 5-28)
PTP[3:0] peak-to-peak jitter, bandwidth between 10 MHz and 80 MHz				200	ps	(Note 5-28)

Note 5-26 Frequency is programmable.

Note 5-27 Both high and low periods are configured to the identical value using PTP:PTP_PINS[0-4]:PIN_WF_HIGH_PERIOD and PTP:PTP_PINS[0-4]:PIN_WF_LOW_PERIOD.

Note 5-28 Timing corrections performed by register writes and high/low periods that do not match a fixed number of core clock cycles will create jitter up to half a core clock cycles.

5.7 Clock Circuit

5.7.1 XI/XO CRYSTAL CHARACTERISTICS

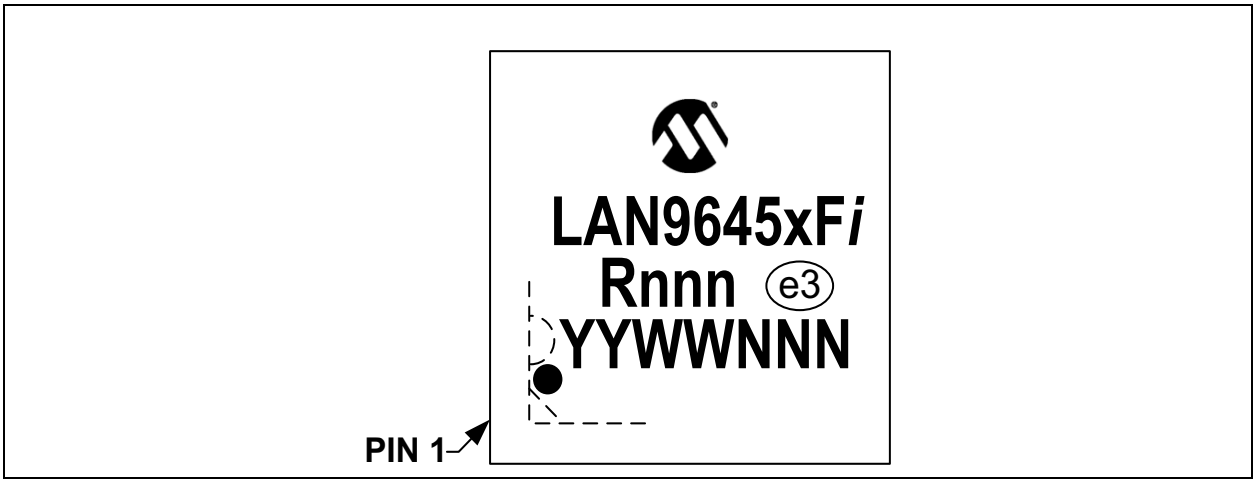
TABLE 5-27: XI/XO CLOCK SPECIFICATIONS

Parameter	Symbol	Min	Typ	Max	Units
Fundamental Frequency	f_0	-100 ppm	25	+2000 ppm	MHz
Load Capacitance	$C_{CRYSTAL}$	6		12.5	pF
Shunt Capacitance	C_{SHUNT}			3	pF
Equivalent Series Resistance	ESR			100	Ω
Motional Capacitance	C_M	1.3		3.2	fF
Drive Level	P_{ON}	300			µW

6.0 PACKAGE INFORMATION

6.1 Top Marking

6.1.1 128-PIN TQFP PACKAGE OPTION TOP MARKING

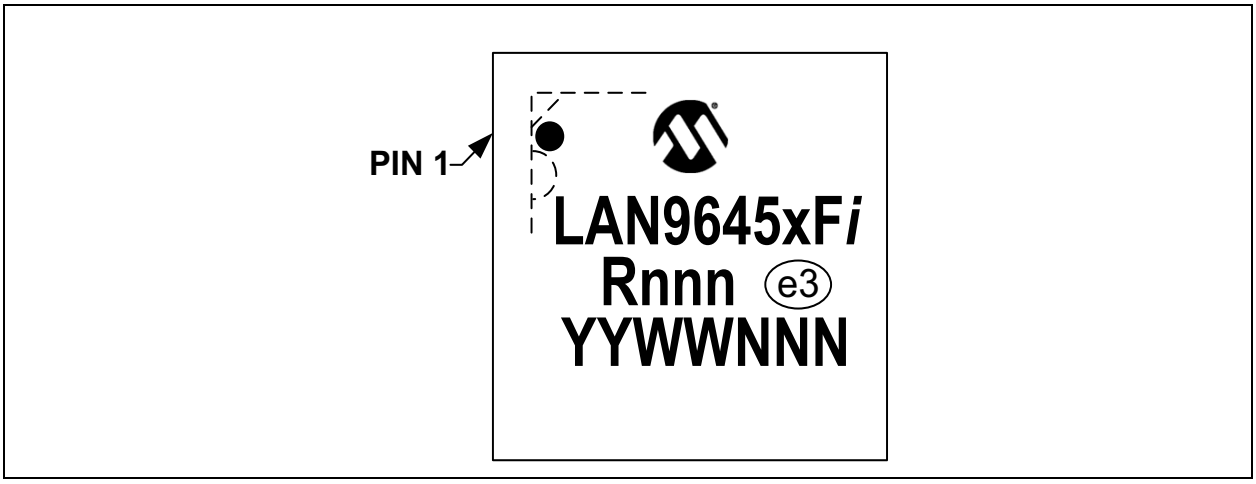


Legend:

x	Part number (5, 7, or 9)
i	Temperature range designator (*i* = industrial)
R	Product revision
nnn	Internal code
e3	Pb-free JEDEC® designator for Matte Tin (Sn)
YY	Year code (last two digits of calendar year)
WW	Week code (week of January 1 is week ‘01’)
NNN	Alphanumeric traceability code

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information

6.1.2 156-PIN VQFN-DR PACKAGE OPTION TOP MARKING



Legend:

x	Part number (5, 7, or 9)
i	Temperature range designator (*i* = industrial)
R	Product revision
nnn	Internal code
e3	Pb-free JEDEC® designator for Matte Tin (Sn)
YY	Year code (last two digits of calendar year)
WW	Week code (week of January 1 is week ‘01’)
NNN	Alphanumeric traceability code

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information

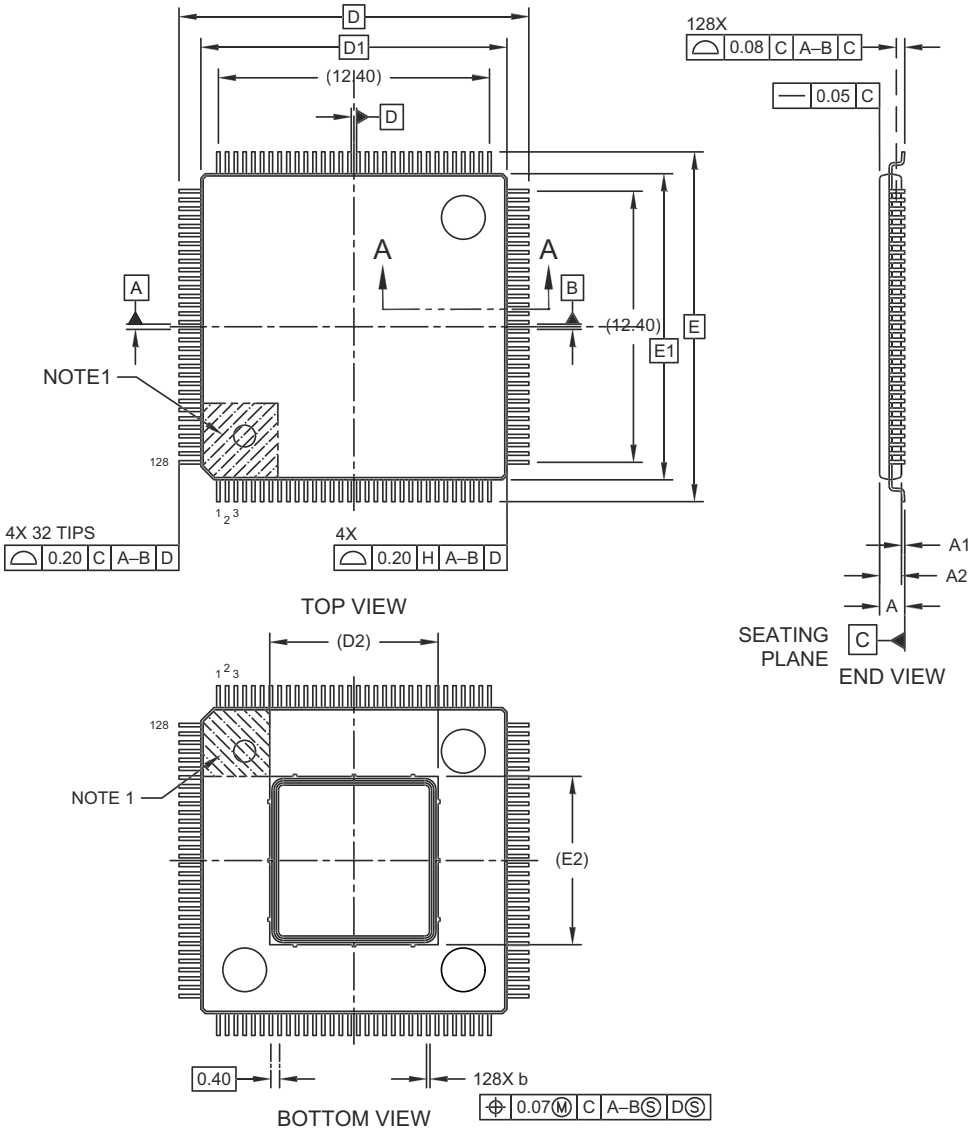
6.2 128-Pin TQFP Package Option

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

FIGURE 6-1: 128-PIN TQFP PACKAGE - DRAWING

128-Lead Plastic Thin Quad Flat Pack (8MW) - 14x14x1.0 mm Body [TQFP]
With 7.7x7.7 mm Grooved Exposed Pad

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



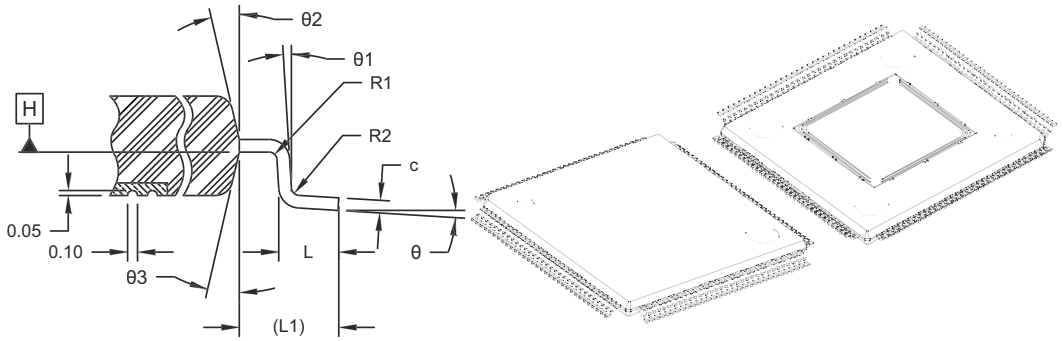
Microchip Technology Drawing C04-00666 Rev A Sheet 1 of 2

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

FIGURE 6-2: 128-PIN TQFP PACKAGE - DIMENSIONS

128-Lead Plastic Thin Quad Flat Pack (8MW) - 14x14x1.0 mm Body [TQFP]
With 7.7x7.7 mm Grooved Exposed Pad

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



SECTION A–A

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		128	
Pitch	e		0.40 BSC	
Overall Height	A	1.00	1.10	1.20
Standoff	A1	0.05	0.10	0.15
Molded Package Thickness	A2	0.95	1.00	1.05
Overall Length	D		16.00 BSC	
Molded Package Length	D1		14.00 BSC	
Overall Width	E		16.00 BSC	
Molded Package Width	E1		14.00 BSC	
Exposed Pad Width	D2	7.60	7.70	7.80
Exposed Pad Length	E2	7.60	7.70	7.80
Terminal Width	b	0.13	0.16	0.23
Terminal Thickness	c	0.09	0.127	0.20
Terminal Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Lead Bend Radius	R1	0.08	-	-
Lead Bend Radius	R2	0.08	-	0.20
Foot Angle	θ	0°	3.5°	7°
Lead Angle	θ1	0°	-	-
Mold Draft Angle	θ2	11°	12°	13°
Mold Draft Angle	θ3	11°	12°	13°

Notes:

1. The Pin 1 visual index feature may vary, but it must be located within the hatched area.
2. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerances, for information purposes only.

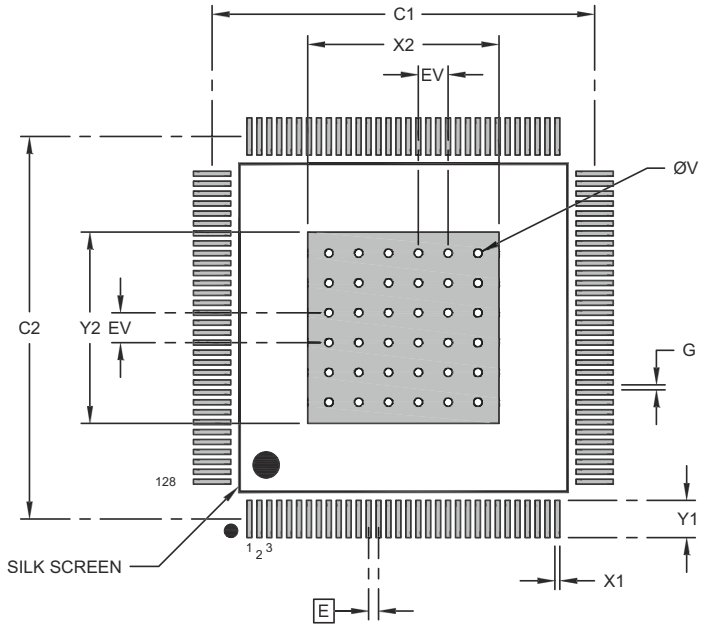
Microchip Technology Drawing C04-00666 Rev A Sheet 2 of 2

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

FIGURE 6-3: 128-PIN TQFP PACKAGE - LAND PATTERN

128-Lead Plastic Thin Quad Flat Pack (8MW) - 14x14x1.0 mm Body [TQFP]
With 7.7x7.7 mm Grooved Exposed Pad

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.40 BSC	
Center Pad Width	X2			7.70
Center Pad Length	Y2			7.70
Contact Pad Spacing	C1		15.40	
Contact Pad Spacing	C2		15.40	
Contact Pad Width (X128)	X1			0.20
Contact Pad Length (X128)	Y1			1.50
Contact Pad to Contact Pad (X124)	G	0.20		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during the reflow process.

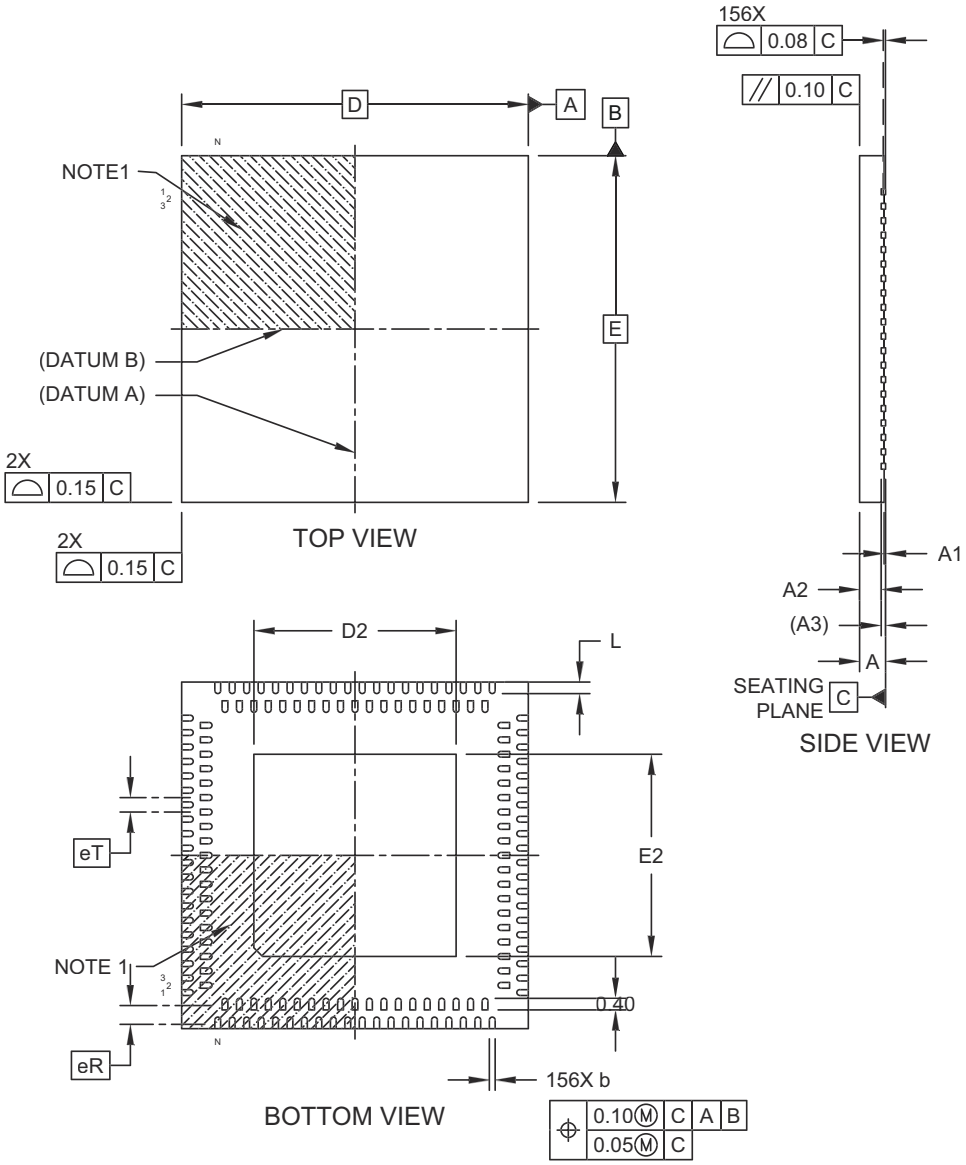
Microchip Technology Drawing C04-02666 Rev A

6.3 156-Pin VQFN-DR Package Option

FIGURE 6-4: 156-PIN VQFN-DR PACKAGE - DRAWING

156-Lead Very-Thin Quad Flatpack No-Leads (8NW) Dual Ring - 12x12x0.9mm Body [VQFN] With 7x7mm Exposed Pad

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



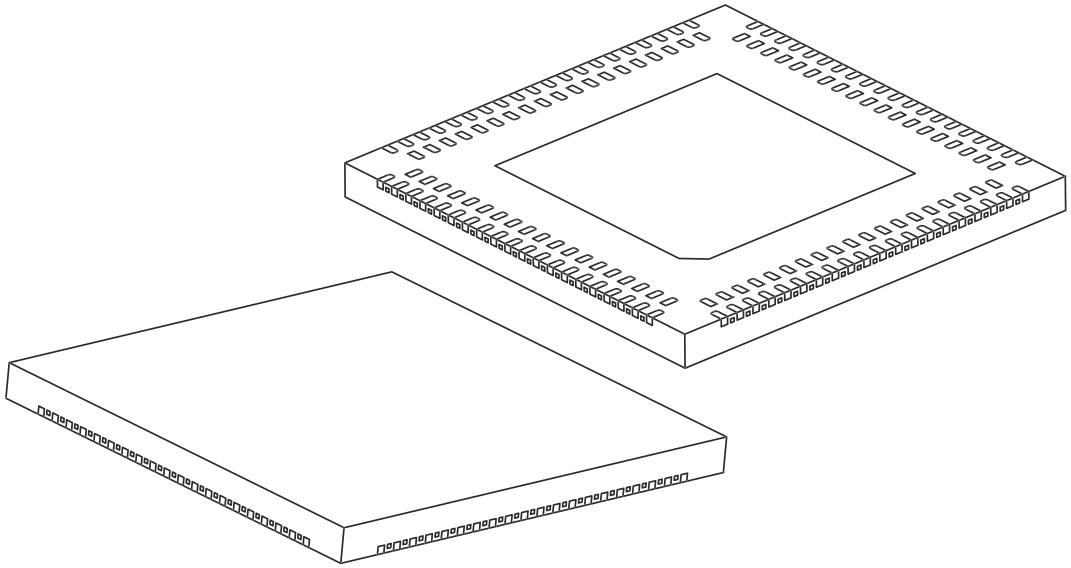
Microchip Technology Drawing C04-00667 Rev B Sheet 1 of 2

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

FIGURE 6-5: 156-PIN VQFN-DR PACKAGE - DIMENSIONS

156-Lead Very-Thin Quad Flatpack No-Leads (8NW) Dual Ring - 12x12x0.9mm Body [VQFN]
With 7x7mm Exposed Pad

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		156	
Pitch of Lead Fingers	eT		0.50 BSC	
Pitch of Lead Fingers Rows	eR		0.65 BSC	
Overall Height	A	0.80	0.85	0.90
Standoff	A1	0.00	0.02	0.05
Mold Thickness	A2	0.65	0.70	0.75
Terminal Thickness	A3		0.152 REF	
Overall Length	D		12.00 BSC	
Exposed Pad Length	D2	6.90	7.00	7.10
Overall Width	E		12.00 BSC	
Exposed Pad Width	E2	6.90	7.00	7.10
Terminal Width	b	0.16	0.20	0.28
Terminal Length	L	0.30	0.40	0.50
Terminal-to-Exposed-Pad	K		1.45 REF	

Notes:

1. The Pin 1 visual index feature may vary, but it must be located within the hatched area.
2. The package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerances, for information purposes only.

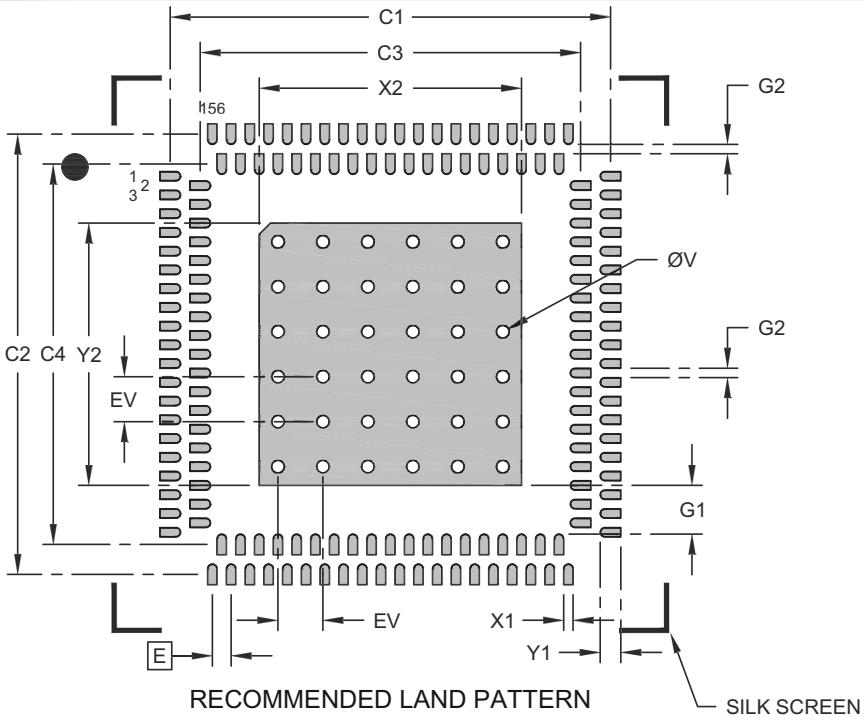
Microchip Technology Drawing C04-00667 Rev B Sheet 2 of 2

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

FIGURE 6-6: 156-PIN VQFN-DR PACKAGE - LAND PATTERN

156-Lead Very-Thin Quad Flatpack No-Leads (8NW) Dual Ring - 12x12x0.9mm Body [VQFN] With 7x7mm Exposed Pad

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Center Pad Width	X2			7.00
Center Pad Length	Y2			7.00
Contact Pad Spacing (Outer Row)	C1		11.75	
Contact Pad Spacing (Outer Row)	C2		11.75	
Contact Pad Spacing (Inner Row)	C3		10.15	
Contact Pad Spacing (Inner Row)	C4		10.15	
Contact Pad Width (X156)	X1			0.25
Contact Pad Length (X156)	Y1			0.55
Contact Pad to Center Pad (X76)	G1	1.30		
Contact Pad to Contact Pad (X148)	G2	0.25		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, pleease refer to current industry standard IPC-7093.

Microchip Technology Drawing C04-02667 Rev B

APPENDIX A: REVISION HISTORY

TABLE A-1: REVISION HISTORY

Revision	Section/Figure/Entry	Correction
DS0006065B (10-31-25)	Throughout document	Confidential removed from footer
	Figure 2-1	SGMII #0 and SGMII #1 TX/RX signal numbering has been revised. SGMII #0 signal names “S1_TXN/P” and “S1_RXN/P” are revised to “S0_TXN/P” and “S0_RXN/P”. SGMII #1 signal names “S2_TXN/P” and “S2_RXN/P” are revised to “S1_TXN/P” and “S1_RXN/P”.
	Figure 3-1, "128-TQFP Pin Assignments (Top View)"	Renamed pins 44 and 52
	Table 3-1, "128-Pin TQFP Assignments"	Renamed pins 44 and 52
	Figure 3-2, "156-Pin VQFN-DR Assignments (Top View)"	Renamed pins 55 and 61
	Table 3-2, "156-Pin VQFN-DR Assignments"	Renamed pins 55 and 61
	Table 3-3, "Pin Assignments", Section 5.2, Operating Conditions**	- Supply voltage has been updated from +1.1V to +1.15V for the following pins: VDD_ALS0, VDD_ALS1, VDD_PLL, VDD_AL, and VDD. - Added note in description of +2.5/3.3V Ethernet PHY and SerDEes Analog High Power Supply - Added the following row: +2.5/3.3V SerDEes Analog High Power Supply
	Table 3-9, "GPIO Alternate Function Descriptions"	Numbering of RGMII and RMII in description for RGMII[1:0]_RXC-REFCLK updated from “RGMII 1/2” and “RMII 1/2” to “RGMII 0/1” and “RMIII 0/1”, respectively.
	Table 4-8, "Receive Counters in the Statistics Block"	Description for short name “c_rx_frag” updated from “Number of short frames with invalid CRC (<64 bytes)” to “Half-duplex links: Number of short frames with invalid CRC (<64 bytes). Full-duplex links: Number of frames received after port is paused. See SYS::FG:MAC_FC_CFG.FC_LATENCY_CFG for details. Note, counter increments if pause frames are received from link partner.”
	Table 4-42, "Port Timestamping Registers"	Changed the following note under this table: Note: The PTP timestamping accuracy on 10Mbps RGMII ports is up to 4 us. to: Note: The PTP timestamping accuracy on 10Mbps RMII/RGMII ports is up to 4 us.

TABLE A-1: REVISION HISTORY (CONTINUED)

Revision	Section/Figure/Entry	Correction
	Section 4.11.6, "Per-Stream Filtering and Policing", Figure 4-13, "PSFP Parameters and Tables", Section 4.11.7, "Stream Gates"	The following total values have been updated: Global stream filters updated from “256” to “128”. Stream gate instances updated from “256” to “128”. Police instances updated from “345 to “226”. Policers for PSFP processing updated from “256 to “128”.
	Section 4.11.8.2, "Stream Table",	Stream table support number updated from “256” to “128”.
	Table 4-29, "Sequence Generation Parameters"	Width for INPUT_PORT_MASK updated from “9” to “10”.
	Section 4.11.8.4, "Individual and Sequence Recovery Functions"	Individual recovery function total value updated from “512” to “256”. Sequence recovery function total value updated from “256” to “128”.
	Table 4-40, "CPU Extraction Header" and Table 4-41, "CPU Injection Header"	The description of field REW_CMD was updated to the following: “Rewriter command, bits 9:0.” The description of field DSCP was updated to the following: “Classified DSCP value. For gPTP frames, this field contains rewriter command, bits 15:10.”
	Section 4.20.1, "Timing Overview"	The following note was removed from the end of the sixth paragraph: “Note, for ports set to a speed of 10 Mbps, hardware timestamping is only supported on SGMII or QSGMII interfaces.”
	Section 4.20.2, "Port Timestamping Configuration"	New section added.
	Section 4.20.6, "Timing Action"	The following text has been updated from: “The timing action is described in a 10-bit vector (16 bit) coming from the VCAP IS2 lookup result REW_CMD. This format is described in Table 4-46. Each 1588 frame passing through the analyzer will be matched against the VCAP rules and processed according to the role of the device and ports involved. Bits 15:10 are described in the gPTP subsection.” to: “The timing action is described in a 6-bit vector coming from the VCAP IS2 lookup result REW_CMD. This format is described in Table 4-46. Each 1588 frame passing through the analyzer will be matched against the VCAP rules and processed according to the role of the device and ports involved.”
	Table 4-47, "REW_CMD Features"	“RewCmd Encoding” addresses updated.
	Section 4.20.13, "gPTP Server Table"	Added two clarifying references to REW_CMD.
	Section 4.22, "Clocking and Reset"	First two bullets of this section have been revised for clarity.

TABLE A-1: REVISION HISTORY (CONTINUED)

Revision	Section/Figure/Entry	Correction
	Section 4.23, "VCore System and CPU Interfaces"	New section added.
	Section 5.1, Absolute Maximum Ratings*	The absolute maximum rating has been updated from 1.21V to 1.265V for the following pins: VDD_ALS0, VDD_ALS1, VDD_PLL, VDD_AL, and VDD.
	Section 5.2, Operating Conditions**	The following sentence was added to the double-asterisked advisory: “Customers are advised to ensure the maximum junction temperature as specified above is not exceeded when choosing the system power level.”
	Table 5-12, "SerDes Transmitter Timing Specifications", Table 5-14, "QSGMII Transmitter Specifications"	- Notes related to slew rate have been removed from below these tables. - Added VDD_AHS to Supply Voltage
	Section 5.4, Current and Power Consumption	Tables with new data added.
	Table 5-3, "Operating Current", Table 5-6, "DC Electrical Characteristics", Table 5-16, "nRESET Timing Specifications", Table 5-17, "MIIM Timing Specifications", Table 5-18, "MIIM Client Timing Specifications", Table 5-20, "SI Timing Specifications", Table 5-21, "SI Client Timing Specifications", Table 5-23, "Serial I/O Timing Specifications", Table 5-26, "Recovered Clock Output Timing Specifications"	Values added and/or updated.
	Table 5-4, "Power Consumption (VDD_AH/VDD_AHS/VDD_IO_x @ 3.3V)", Table 5-5, "Power Consumption (VDD_AH/VDD_AHS/VDD_IO_x @ 2.5V)"	Updated table titles by adding /VDD_AHS
	Figure 5-3, "MIIM CLIENT TIMING DIAGRAM"	Added Figure 5-3, "MIIM CLIENT TIMING DIAGRAM"
	Figure 5-8, "SERIAL I/O TIMING"	Added Figure 5-8, "SERIAL I/O TIMING"
	Section 6.1, Top Marking, Product Identification System	Top Marking and Product Ordering Codes have been updated to include industrial temperature range designator.
DS0006065A (07-10-25)	Initial Release.	

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X](1)	- I	/	XXX
Device	Tape and Reel Option			Package

Device:	LAN96459F	= 9-port switch with 5 Integrated PHYs and 4 configurable interfaces
	LAN96457F	= 7-port switch with 5 Integrated PHYs and 4 configurable interfaces
	LAN96455F	= 5-port switch with 5 Integrated PHYs and 4 configurable interfaces
Temperature:	I	= -40°C to +85°C (Industrial)
Tape and Reel Option:	Blank	= Standard packaging (tray)
	T	= Tape and Reel (Note 1)
Package:	8MW	= 128-pin TQFP
	8NW	= 156-pin VQFN-DR

Examples:

a) LAN96459F-I/8MW
9-port switch, Tray, 128-pin TQFP

b) LAN96457FT-I/8MW
7-port switch, Tape & Reel, 128-pin TQFP

c) LAN96459FT-I/8MW
9-port switch, Tape & Reel, 128-pin TQFP

d) LAN96459F-I/8NW
9-port switch, Tray, 156-pin VQFN-DR

e) LAN96457FT-I/8NW
7-port switch, Tape & reel, 156-pin VQFN-DR

f) LAN96455FT-I/8NW
5-port switch, Tape & reel, 156-pin VQFN-DR

Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legalinformation/microchip-trademarks.

ISBN: 979-8-3371-2220-5

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.