VSC7513

8-Port L2 Gigabit Ethernet Switch

INTRODUCTION

The VSC7513 Industrial IoT Ethernet switch contains four integrated dual media 10/100/1000BASE-T PHYs, two 1G SGMII/SerDes, and an option for either a 1G/2.5G SGMII/SerDes Node Processor Interface (NPI) or a PCIe interface for external CPU connectivity. The NPI/PCIe can operate as a standard Ethernet port. The device provides a rich set of Industrial Ethernet switching features such as fast protection switching, 1588 precision time protocol, and synchronous Ethernet. Advanced TCAM-based VLAN and QoS processing enable delivery of differentiated services. Security is assured through frame processing using Microsemi's TCAM-based Versatile Content Aware Processor. In addition, the device contains a powerful 500 MHz CPU, enabling full management of the switch.

The device supports the following main port configurations.

TABLE 1: MAIN PORT CONFIGURATIONS

Ports	1G Dual Media	1G SGMII	QSGMII	2.5G NPI	PCIe
6 + NPI	4	2	—	1	—
6 + PCIe	4	2	—	—	1
8 + NPI	4	—	1	1	—
8 + PCIe	4	—	1	—	1

General Features

- 1.75 megabits of integrated shared packet memory
- 4 × integrated dual media copper PHYs/SGMII ports
- Fully nonblocking wire-speed switching performance with weighted random early detection (WRED) for all frame sizes
- Eight QoS classes and eight queues per port with strict or deficit weighted round robin scheduling
- Dual leaky bucket policers, per QoS class and per port
- Dual leaky bucket policers, flow-based through TCAM matching
- DWRR and strict priority egress scheduler/shaper and 9 dual leaky bucket shapers per egress port.
- TCAM-based classification entries for Quality of Service (QoS) and VLAN membership
- TCAM-based host identity entries for source IP guarding
- TCAM-based security enforcement entries
- TCAM-based egress tagging entries
- L1 Synchronous Ethernet
- VeriTime™—Microchip's patent-pending distributed timing technology that delivers the industry's most accurate IEEE 1588v2 timing implementation for both one-step and two-step clocks
- Audio/Video bridging (AVB) with support for time-synchronized, low-latency audio and video streaming services
- Energy Efficient Ethernet (IEEE 802.3az)
- VCore-III CPU system with integrated 500 MHz MIPS 24KEc™ CPU with MMU and DDR3 SDRAM controller
- PCIe 1.x and NPI CPU interface
- Internal shared memory buffer (8 queues per port) Integrated 10/100/1000BASE-T Ethernet copper transceiver (IEEE 802.3ab compliant) with the industry's only non-TDR-based VeriPHY™ cable diagnostics algorithm
- Patented line driver with low EMI voltage mode architecture and integrated line-side termination resistors
- Wake-on-LAN using magic packets
- HP Auto-MDIX and manual MDI/MDIX support

LAYER 2 SWITCHING

- 4,096 MAC addresses
- 4,096 VLANs (IEEE 802.1Q)
- Push/pop/translate up to three VLAN tags; translation on ingress and/or on egress
- Policing with storm control and MC/BC protection
- TCAM-based VLAN classification and translation with pattern matching against Layer 2 through Layer 4 information such as MAC addresses, VLAN tag headers, EtherType, DSCP, IP addresses, and TCP/UDP ports and ranges
- QoS and VLAN TCAM entries
- VLAN egress tagging TCAM entries
- Link aggregation (IEEE 802.3ad)
- Link aggregation traffic distribution is programmable and based on Layer 2 through Layer 4 information
- Wire-speed hardware-based learning and CPU-based learning configurable per port
- Independent and shared VLAN learning
- Provider Bridging (VLAN Q-in-Q) support (IEEE 802.1ad)
- Rapid Spanning Tree Protocol support (IEEE 802.1w)
- Multiple Spanning Tree Protocol support (IEEE 802.1s)
- Jumbo frame support
- TCAM-based classification and VCAP-II security

LAYER 2 MULTICAST

- IPv4/IPv6 multicast Layer-2 switching with up to 4,096 groups and 64 port masks
- Internet Group Management Protocol version 2 (IGMPv2) support
- Internet Group Management Protocol version 3 (IGMPv3) support with source specific multicast forwarding
- Multicast Listener Discovery (MLDv1) support
- Multicast Listener Discovery (MLDv2) support with source specific forwarding

QUALITY OF SERVICE

- Eight QoS classes and two drop precedence levels
- Eight QoS queues per port with Deficit Weighted Round-robin (DWRR) or Frame Based Round-robin (FBRR) scheduling
- TCAM-based QoS and VLAN classification with pattern matching against Layer 2 through Layer 4 information
- DSCP translation, both ingress and/or egress
- DSCP remarking based on QoS class and drop precedence level
- VLAN (PCP, DEI, and VID) translation, both ingress and egress
- PCP and DEI remarking based on QoS class and drop precedence level
- Policers, selectable per QoS class, per queue, per port, and per security entry through TCAM-based pattern matching, programmable in steps of 33.3 kbps
- Dual leaky bucket shaping per port and per QoS class, programmable in steps of 100 kbps
- Full-duplex flow control (IEEE 802.3X) and half-duplex backpressure, symmetric and asymmetric
- Priority-based full-duplex flow control (IEEE 802.1Qbb)
- Multicast and broadcast storm control with flooding control
- QoS classification based on IEEE 802.1p and IPv4/IPv6 DSCP

SECURITY

- Versatile Content Aware Processor (VCAP) packet filtering engine using ACLs for ingress and egress packet inspection
- Security VCAP entries
- Source IP guarding entries
- Shared VCAP rate policers with rate measurements in frames per second or bits per second
- Eight shared range checkers supporting ranges based on TCP/UDP port numbers, DSCP values, and VLAN identifiers

- VCAP match patterns supporting generic MAC, ARP, IPv4, and IPv6 protocols
- VCAP actions including permit/deny, police, count, CPU-copy, and mirror
- Special support for IP fragments, UDP/TCP port ranges, and ARP sanity check
- Extensive CPU DoS prevention by VCAP rate policers and hit-me-once functions
- Surveillance functions supported by 32-bit VCAP counters
- Generic storm controllers for flooded broadcast, flooded multicast, and flooded unicast traffic
- Selectable CPU extraction queues for segregation of CPU redirected traffic, with 8 extraction queues supported
- Per-port, per-address registration for copying/redirecting/discarding of reserved IEEE MAC addresses (BPDU, GARP, CCM/Link trace)
- Port-based and MAC-based access control (IEEE 802.1X)
- Per-port CPU-based learning with option for secure CPU-based learning
- Per-port ingress and egress mirroring
- Mirroring per VLAN and per VCAP match

MANAGEMENT

- VCore-III™ CPU system with integrated 500 MHz MIPS 24KEc CPU with MMU and DDR3/DDR3L SDRAM controller
- PCIe 1.x CPU interface
- CPU frame extraction (eight queues) and injection (two queues) through DMA, which enables efficient data transfer between Ethernet ports and CPU/PCIe
- EJTAG debug interface
- Configurable 16-bit or 8-bit DDR3 SDRAM interface supporting up to one gigabyte (GB) of memory
- Twenty-two pin-shared general-purpose I/Os:
- Serial GPIO and LED controller controlling up to 32 ports with four LEDs each
- Dual PHY management controller (MIIM)
- Dual UART
  Built-in two wire serial interface multiplexer
- External interrupts
- 1588 synchronization I/Os
- SFP loss of signal inputs
- External access to registers through PCIe, SPI, MIIM, or through an Ethernet port with inline Microchip's Versatile Register Access Protocol (VRAP)
- Per-port counter set with support for the RMON statistics group (RFC 2819) and SNMP interfaces group (RFC 2863)
- Energy Efficient Ethernet (EEE) (IEEE 802.3az)
- Synchronous Ethernet, with two clock outputs recovered from any port
- Support for CPU modes with internal CPU only, external CPU only, or dual CPU

Product Parameters

All SerDes, copper transceivers, packet memory, and configuration tables necessary to support network applications are integrated. The following table lists the primary parameters for the device.

TABLE 2: PRODUCT PARAMETERS

Features and Port Configurations	VSC7513
Maximum I/O bandwidth excluding 2.5G NPI port	8 Gbps
Maximum number of ports	8 + NPI 8 + PCIe
Maximum number of 2.5G SGMII ports including NPI port	1
Maximum number of 1G SGMII ports including NPI port	7
Maximum number of 1G dual media ports (1G SGMII or 1G CuPHY)	4
Maximum number of QSGMII ports	1

TABLE 2: PRODUCT PARAMETERS (CONTINUED)

Features and Port Configurations	VSC7513
2.5G SGMII NPI port (also capable of 1G SGMII or PCIe)	1
CuPHYs	4
SERDES1G lanes	6
SERDEDS6G lanes including 2.5G NPI	2
Layer 2 Switching	
Packet buffer	1.75 Mbits
MAC table size	4K
VLAN table size	4K
Layer 2 multicast port masks	64
Quality of Service and Security	
VCAP IS1 entries	64 full, or 128 half, or 256 quad
VCAP IS2 entries	64 full, or 128 half, or 256 quad
VCAP ES0 entries	256

Applications

VSC7513 targets unmanaged and managed applications in industrial Ethernet, Internet of Things (IoT), Enterprise, SMB, Customer-premises Equipment (CPE), and Network Termination Equipment (NTE).

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.

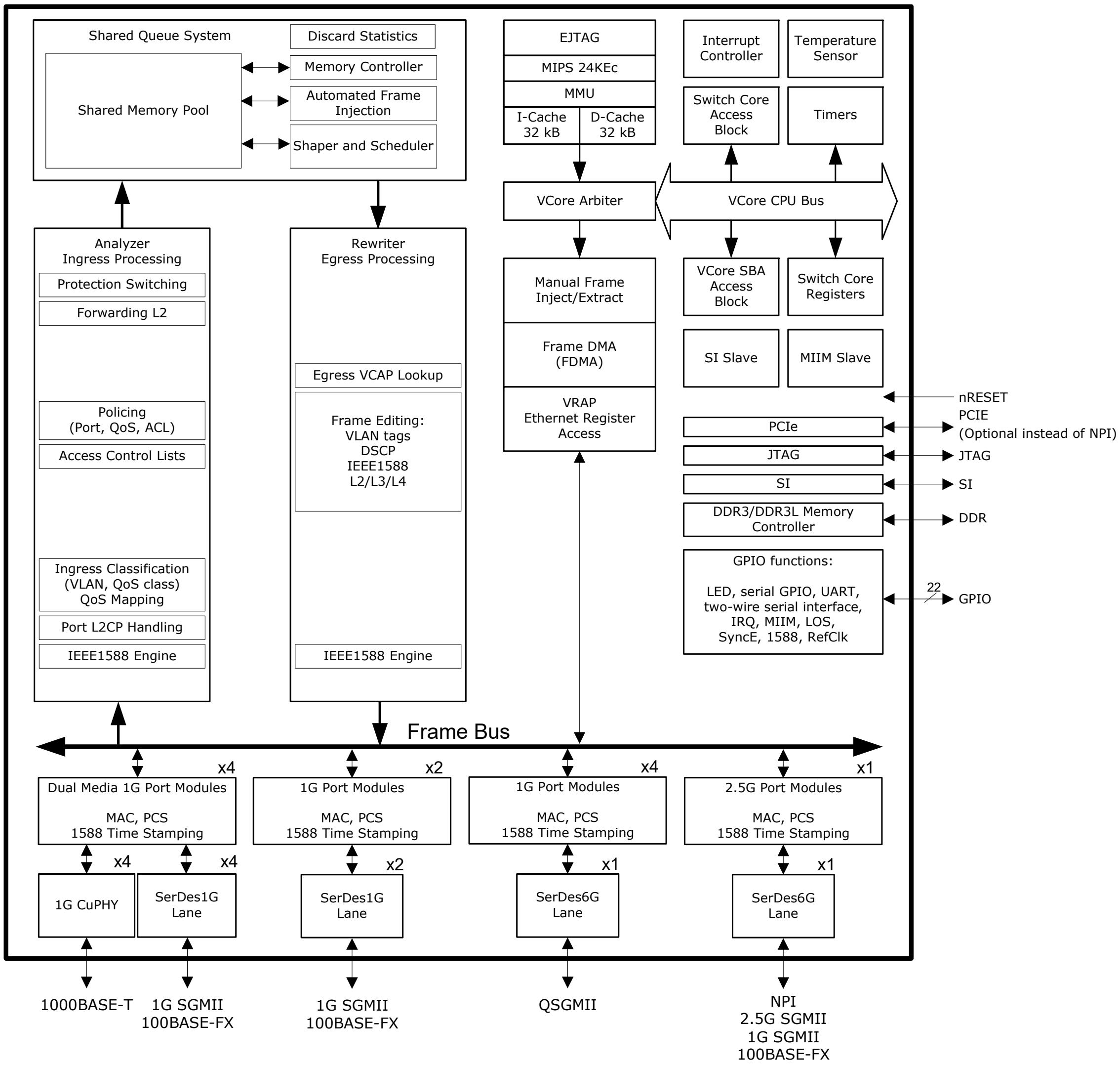
Contents

1.0 FUNCTIONAL OVERVIEW

This section provides an overview of all major blocks and functions involved in the forwarding operation in the same order as a frame traverses through the VSC7513 device. It also outlines other major functionality of the device such as the CPU port module, the CPU system, and CPU interfaces.

The following illustration is a block diagram for the device.

FIGURE 1-1: BLOCK DIAGRAM



1.1 Frame Arrival

The Ethernet interfaces receive incoming frames and forwards these to the port modules. The 1G SGMII and 2.5G SGMII ports support both 100BASE-X and 1000BASE-X-SERDES.

Each port module contains a Media Access Controller (MAC) that performs a full suite of checks, such as VLAN Tag aware frame size checking, Frame Check Sequence (FCS) checking, and Pause frame identification.

Each port module that connects to a SerDes block contains a Physical Coding Sublayer (PCS), which perform 8 bits/ 10 bits encoding, auto-negotiation of link speed and duplex mode, and monitoring of the link status.

Full-duplex is supported for all speeds, and half-duplex is supported for 10 Mbps and 100 Mbps. Symmetric and asymmetric pause flow control are both supported as well as priority-based flow control (IEEE 802.1Qbb).

All Ethernet ports support Energy Efficient Ethernet (EEE) according to IEEE 802.3az. The shared queue system is capable of controlling the operating states, active or low-power, of the PCS. The PCS understands the line signaling as required for EEE. This includes signaling of active, sleep, quiet, refresh, and wake.

1.2 Basic and Advanced Frame Classification

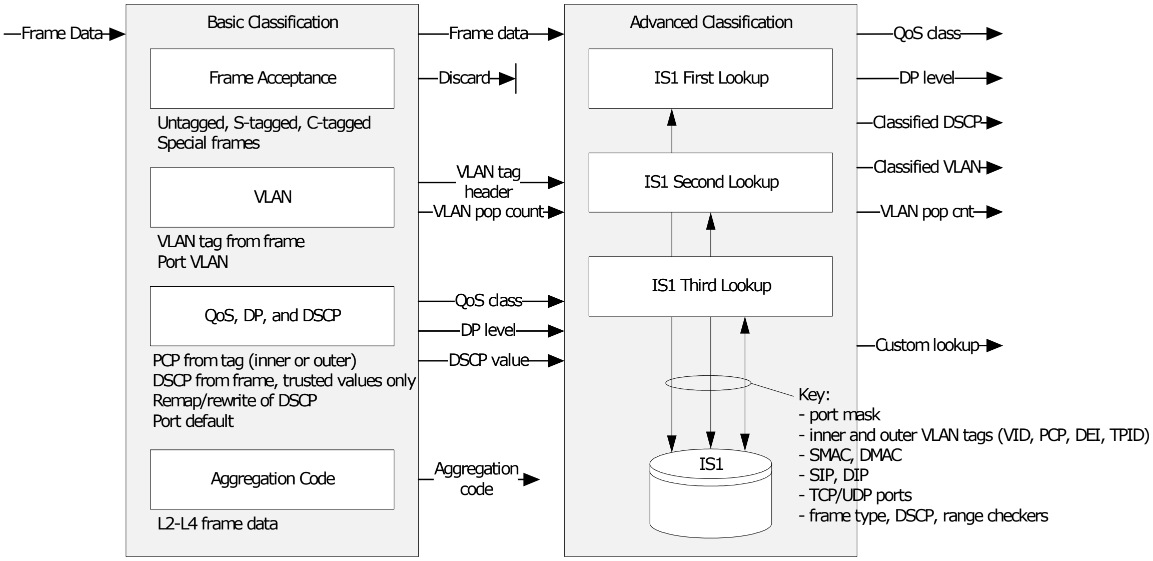
The basic and advanced frame classification in the ingress processing module receive all frames. The basic and advanced classifiers determine a range of frame properties such as VLAN, Quality of Service (QoS) class, and drop precedence level. This information is carried through the switch together with the frame and affects policing, drop precedence marking, statistics collecting, security enforcement, Layer-2 forwarding, and rewriting.

The classification is a combination of a basic classification using configurable logic and more advanced classification using a TCAM.

The classification engine understands up to two VLAN tags and can look for Layer-3 and Layer-4 information behind two VLAN tags. If frames are triple tagged, the higher-layer protocol information is not extracted.

The following illustration shows the basic and advanced frame classification.

FIGURE 1-2: BASIC AND ADVANCED FRAME CLASSIFICATION



The basic classification classifies each frame to a VLAN, a QoS class, a drop precedence (DP) level, DSCP value, and an aggregation code. The basic classification also performs a general frame acceptance check. The output from the basic classification may be overwritten or changed by the more intelligent advanced classification using the IS1 TCAM.

Frame Acceptance The frame acceptance filter checks for valid combinations of VLAN tags against the ingress port's VLAN acceptance filter where it is possible to configure rules for accepting untagged, priority-tagged, C-, and S-tagged frames. In addition, the filter also enables discarding of frames with illegal MAC addresses (for instance null MAC address or multicast source MAC address).

VLAN Every incoming frame is classified to a VLAN by the basic VLAN classification. This is based on the VLAN in the frame, or if the frame is untagged or the ingress port is VLAN unaware, it is based on the ingress port's default VLAN. A VLAN classification includes the whole TCI (PCP, DEI, and VID) and also the TPID (C-tag or S-tag).

For double-tagged frames, it is selectable whether the inner or the outer tag is used.

The device can recognize S-tagged frames with the standard TPID (0x88A8) or S-tagged frames using a custom programmable value. One custom value is supported by the device.

QoS, DP, and DSCP Each frame is classified to a Quality of Service (QoS) class and a drop precedence level (frame color: green/yellow). The QoS class and DP level are used throughout the device for providing queuing, scheduling, and congestion control guarantees to the frame according to what is configured for that specific QoS class and color.

The QoS class and DP level in the basic classification are based on the class of service information in the frame's VLAN tags (PCP and DEI) and/or the DSCP values from the IP header. Both IPv4 and IPv6 are supported. If the frame is non-IP or untagged, the port's default QoS class and DP level are used.

The DSCP values can be remapped before being used for QoS. This is done using a common table mapping the incoming DSCP to a new value. Remapping is enabled per port. In addition, for each DSCP value, it is possible to specify whether the value is trusted for QoS purposes.

Each IP frame is also classified to an internal DSCP value. By default, this value is taken from the IP header but it may be remapped using the common DSCP mapping table or rewritten based on the assigned QoS class. The classified DSCP value may be written into the frame at egress – this is programmable in the rewriter.

Aggregation Code The basic classification calculates an aggregation code, which is used to select between ports that are member of a link aggregation group. The aggregation code is based on selected Layer-2 through Layer-4 information, such as MAC addresses, IP addresses, IPv6 flow label, ISDX, and TCP/UDP port numbers. The aggregation code ensures that frames belonging to the same conversation are using the same physical ports in a link aggregation group.

1.2.1 ADVANCED CLASSIFICATION

Following basic classification, Layer-2 and Layer-4 information is extracted from each frame and matched against a TCAM, IS1, with one of the following six different IS1 keys:

- NORMAL. Up to 128 entries with primary fields in key consisting of SMAC, outer VLAN tag, 32-bit source IP address, IP protocol, TCP/UDP source and destination port
- NORMAL_IP6. Up to 64 entries with primary fields in key consisting of SMAC, inner and outer VLAN tags, 128-bit source IP address, IP protocol, TCP/UDP source and destination port
- 7TUPLE. Up to 64 entries with primary fields in key consisting of source and destination MAC addresses, inner and outer VLAN tags, 64-bit source and destination IP addresses, IP protocol, TCP/UDP source and destination port
- 5TUPLE_IP4. Up to 128 entries with primary fields in key consisting of inner and outer VLAN tags, 32-bit source and destination IP addresses, IP protocol, TCP/UDP source and destination port
- 5TUPLE_IP6. Up to 64 entries with primary fields in key consisting of inner and outer VLAN tags, 128-bit source and destination IP addresses, IP protocol, TCP/UDP source and destination port
- S1_DBL_VID. Up to 256 entries with primary fields in key consisting of inner and outer VLAN tags

The TCAM embeds powerful protocol awareness for well-known protocols such as LLC, SNAP, IPv4, IPv6, and UDP/TCP. For each frame, three keys are generated and matched against the TCAM. The keys are selectable per ingress port per frame type (IPv4, IPv6, non-IP) per IS1 lookup.

The actions associated with each entry (programmed into the TCAM action RAM) include the ability to overwrite or translate the classified VLAN, overwrite the priority code point (PCP) or the drop eligibility indicator (DEI), overwrite the QoS class and DP level, or overwrite the DSCP value. Each of these actions is enabled individually for each of the three lookup.

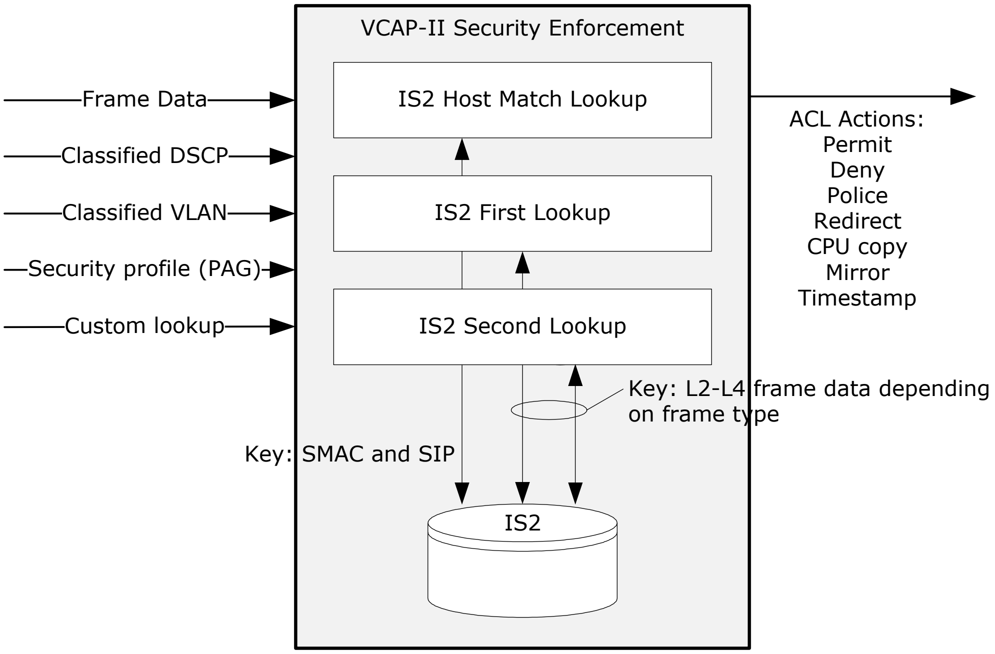
In addition, a policy association group (PAG) is assigned to the frame. The PAG identifies a security profile to which the frame belongs. The PAG is used in the succeeding security frame processor, IS2, to select which access control lists to apply to the frame. The PAG enables creating efficient ACLs that only are applicable to frames with the same PAG.

A lookup in IS1 can enable the use of a custom lookup in IS2, the security enforcement TCAM. The custom lookup can match against selectable frame fields from the incoming frame. In total, 40 bytes from the frame are matched against the TCAM. This is a powerful future-proofing feature enabling handling of new protocols.

1.3 Versatile Content Aware Processor (VCAP)

All frames are inspected by the VCAP IS2 before they are passed on to the Layer-2 forwarding. The following illustration depicts VCAP security enforcement.

FIGURE 1-3: VERSATILE CONTENT AWARE PROCESSOR



The VCAP uses a TCAM-based frame processor enabling implementation of a rich set of security features. The flexible VCAP engine supports wire-speed frame inspection based on Layer 2-4 frame information, including the ability to perform longest prefix matching and identifying port ranges. The action associated with each VCAP entry (programmed into the VCAP action RAM) includes the ability to do frame filtering, dual leaky bucket rate limitation (frame or byte based), snooping to CPU, redirection to CPU, mirroring, 1588 time stamping, and accounting. Even though the VCAP is located in the ingress path of the device, it possesses both ingress and egress capabilities.

The VCAP embeds powerful protocol awareness for well-known protocols such as LLC, SNAP, ARP, IPv4, IPv6, and UDP/TCP. IPv6 is supported with full matching against both the source and the destination IP addresses.

Each frame is looked up three times in IS2. The first lookup is a host match lookup for IPv4 and IPv6 frames enabling MAC/IP binding and IP source guarding. The key consists of information identifying the source host: ingress port number, source MAC address, and full source IP address. The output is an action informing the following two lookups in IS2 of whether the host is accepted into the network. The following two lookups in IS2 construct a key based on the frame type (LLC, SNAP, ARP, IPv4, IPv6, UDP/TCP, OAM, custom) extracting relevant information. IS2 supports up to 256 host match entries, 128 LLC, SNAP, ARP, IPv4-TCP/UDP, or OAM entries and 64 IPv6-TCP/UDP, and custom entries.

1.4 Policing

Each frame is subject to a number of different policing operations. The device features 192 programmable policers. The policers are split into the followings groups:

- Queue policers: ingress port number and QoS class determine which policer to use.
- Port policers: ingress port number determines which policer to use.
- VCAP IS2 policers: an IS2 action can point to a policer.

The internal CPU port is the 12th port and also includes port and queue policers. The policers can measure frame rates or bit rates.

Each frame can trigger up to three policers: a queue policer, a port policer, and a VCAP IS2 policer.

Each policer is a MEF-compliant dual leaky bucket policer supporting both color-blind and color-aware operation. The initial frame color is derived from the drop precedence level from the frame classification. For color-aware operation, a coupling mode is configurable for each policer.

Each frame is counted in statistics reflecting the ingress port, the QoS class, and whether the frame was discarded by one of the policers or not.

Finally, the analyzer contains a group of storm control policers that are capable of policing various kinds of flooding traffic as well as CPU directed learn traffic. These policers are global policers working on all frames received by the switch. Storm policers measure frame rates.

1.5 Layer-2 Forwarding

After the policers, the Layer-2 forwarding block (the analyzer) handles all fundamental forwarding operations and maintains the associated MAC table, the VLAN table, and the aggregation table. The device implements an 4K MAC table and a 4K VLAN table.

The main task of the analyzer is to determine the destination port set of each frame. This forwarding decision is based on various information such as the frame’s ingress port, the source MAC address, the destination MAC address, the VLAN identifier, as well as the frame’s VCAP action, mirroring, and the destination port’s link aggregation configuration.

The switch performs Layer-2 forwarding of frames. For unicast and Layer-2 multicast frames, this means forwarding based on the destination MAC address and the VLAN. For IPv4 and IPv6 multicast frames, the switch performs Layer-2 forwarding, but based on Layer-3 information, such as the source IP address. The latter enables source-specific IPv4 multicast forwarding (IGMPv3) and source-specific IPv6 multicast forwarding (MLDv2).

The following are some of the contributions to the Layer-2 forwarding.

- VLAN classification. VLAN-based forward filtering includes source port filtering, destination port filtering, VLAN mirroring, asymmetric VLANs, and so on.
- Security enforcement. The security decision made by the VCAP can, for example, redirect the frame to the CPU based on some abnormality detection filters.
- MSTP. The VLAN identifier maps to a Multiple Spanning Tree instance, which determines MSTP-based destination port filtering.
- MAC addresses. Destination and source MAC address lookups in the MAC table determine if a frame is a learn frame, a flood frame, a multicast frame, or a unicast frame.
- Learning. By default, the device performs wire-speed learning on all ports. However, certain ports could be configured with secure learning enabled, where an incoming frame with unknown source MAC address is classified as a “learn frame” and is redirected to the CPU. The CPU performs the learning decision and also decides whether the frame is forwarded.
  Learning can also be disabled. In that case, it does not matter if the source MAC address is in the MAC table.
- Link aggregation. A frame targeted at a link aggregate is further processed to determine which of the link aggregate group ports the frame must be forwarded to.
- Mirroring. Mirror probes may be set up in different places in the forwarding path for monitoring purposes. As part of a mirror a copy of the frame is sent either to the CPU or to another port.

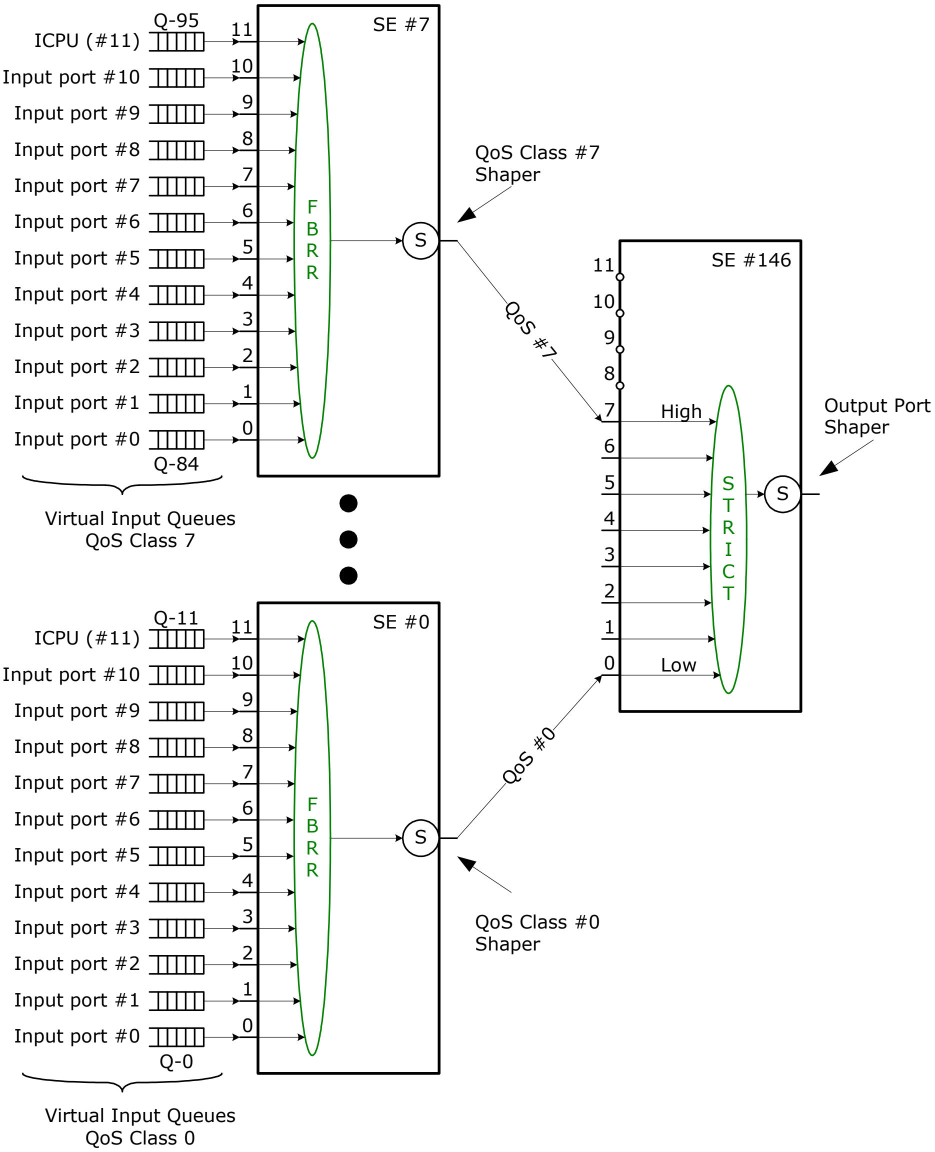
1.6 Shared Queue System and Egress Scheduler

The analyzer provides the destination port set of a frame to the shared queue system. It is the queue system’s task to control the frame forwarding to all destination ports.

The shared queue system embeds 1.75 megabits of memory that can be shared between all queues and ports. The queue system implements egress queues per priority per ingress port. The sharing of resources between queues and ports is controlled by an extensive set of thresholds. The overall frame latency through the switch is low due to the shared queue system storing the frame only once.

Each egress port implements a scheduler and shapers as shown in the following illustration. Per egress port, the scheduler sees the outcome of aggregating the egress queues (one per ingress port per QoS class) into eight QoS classes. The aggregation is done in a frame based round-robin fashion (FBRR) per QoS class serving all ingress ports equally. By default, strict scheduling is performed between QoS classes for the port. QoS class 7 has strict highest priority while QoS class 0 has strict lowest priority.

FIGURE 1-4: DEFAULT EGRESS SCHEDULER AND SHAPER CONFIGURATION

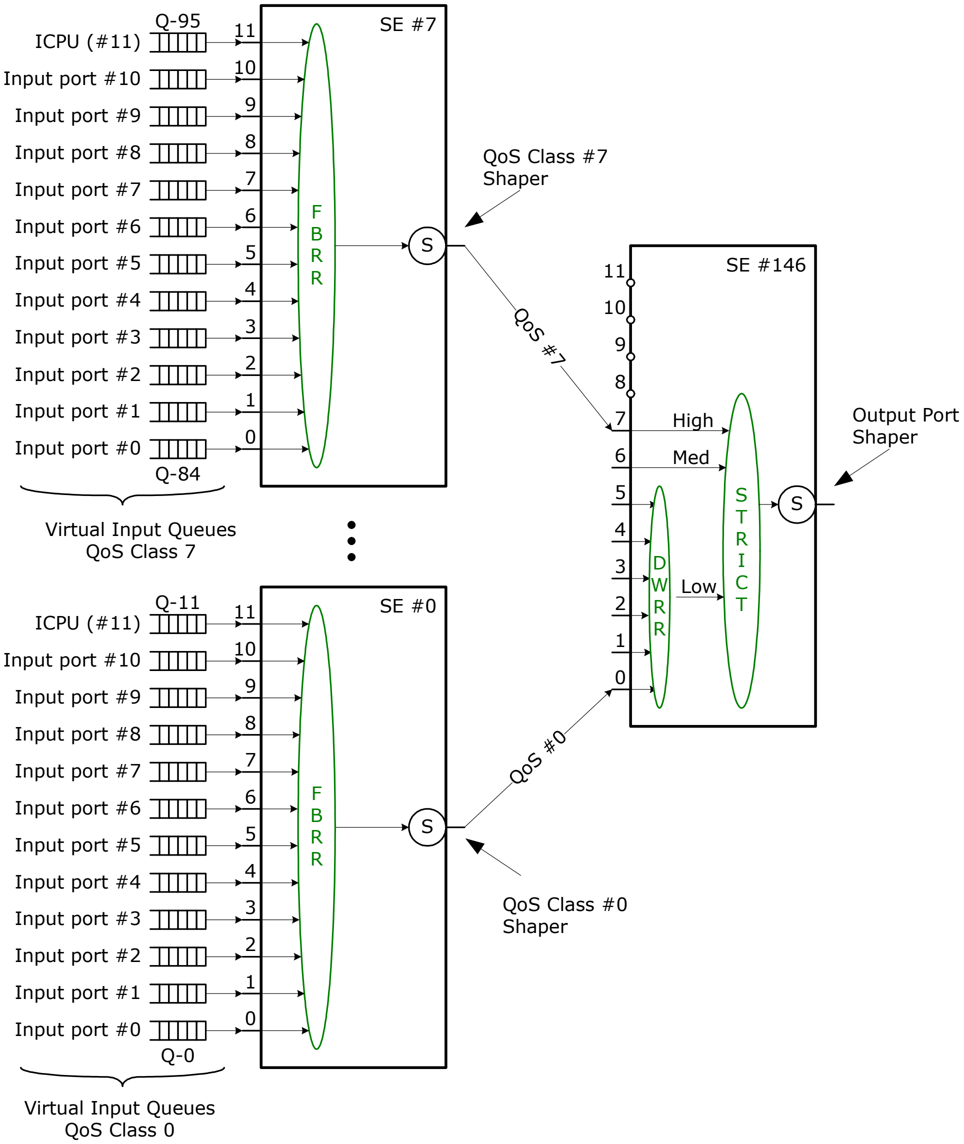


Scheduling between QoS classes within the port can use one of three methods:

- Strict. Frames with the highest QoS class are always transmitted before frames with lower QoS class. This is shown in Figure 1-4, page 13.
- Combination of strict and Deficit Weighted Round Robin (DWRR) scheduling. Any split of strict and DWRR QoS classes can be configured. Figure 1-3, page 11 shows an example where QoS classes 6 and 7 are strict while QoS classes 0 through 5 are weighted. Each QoS class sets a DWRR weight ranging from 0 to 31.
- Combination of strict and Frame Based Round Robin (FBRR) scheduling. Any split between strict and FBRR QoS classes can be configured.

All shapers shown in the following illustrations are single leaky bucket shapers.

FIGURE 1-5: ALTERNATIVE EGRESS SCHEDULER AND SHAPER CONFIGURATION

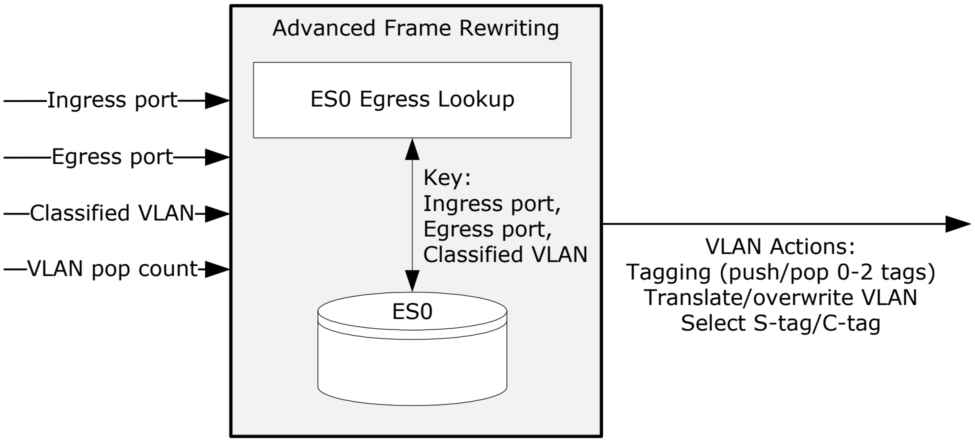


1.7 Rewriter and Frame Departure

Before transmitting the frame on the egress line, the rewriter can modify selected fields in the frame, such as VLAN tags, DSCP value, time stamping, and FCS.

The rewriter controls the final VLAN tagging of frames based on the classified VLAN, the VLAN pop count, and egress-determined VLAN actions from the ES0 TCAM lookup. The egress VLAN actions are by default given by the egress port settings. These include normal VLAN operations such as pushing a VLAN tag, untagging for specific VLANs, and simple translations of DEI and PCP.

FIGURE 1-6: ADVANCED FRAME REWRITING



By using the egress TCAM, ES0, much more advanced VLAN tagging operations can be achieved. ES0 enables pushing up to two VLAN tags and allows for a flexible translation of the VLAN tag header. The key into ES0 is the combination of the ingress port, the egress port, and the classified VLAN tag header.

The PCP and DEI bits in the VLAN tag are subject to remarking based on translating the classified tag header or by using the classified QoS value and the frame's drop precedence level from ingress.

In addition, the DSCP value in IP frames can be updated using the classified DSCP value and the frame's drop precedence level from ingress. The DSCP value can be remapped at egress before writing it into the frame.

Finally, the rewriter updates the FCS if the frame was modified before the frame is transmitted.

The egress port module controls the flow control exchange of pause frames with a neighboring device when the interconnection link operates in full-duplex flow control mode. When the connected device triggers flow control through transmission of a pause frame, the MAC stops the egress scheduler's forwarding of frames out of the port. Traffic then builds up in the queue system, but sufficient queuing is available to ensure wire speed loss-less operation.

In half-duplex operation, the port module's egress path responds to back pressure generation from a connected device by collision detection and frame retransmission.

1.8 CPU Port Module

The CPU port module contains eight CPU extraction queues and two CPU injection queues. These queues provide an interface for exchanging frames between the internal CPU system and the switch core. An external CPU using the serial interface can also inject and extract frames to and from the switch core by using the CPU port module. Any Ethernet interface on the device can also be used for extracting and injecting frames.

The switch core can intercept a variety of different frame types and copy or redirect these to the CPU extraction queues. The classifier can identify a set of well-known frames such as IEEE, reserved destination MAC addresses (BPDUs, GARPs, CCM/Link trace), and IP-specific frames (IGMP, MLD). The security TCAM, IS2, provides another very flexible way of intercepting all kinds of frames, for instance specific OAM frames, ARP frames or explicit applications based on TCP/UDP port numbers. In addition, frames can be intercepted based on the MAC table, the VLAN table, or the learning process.

Whenever a frame is copied or redirected to the CPU, a CPU extraction queue number is associated with the frame and used by the CPU port module when enqueuing the frame into the 8 CPU extraction queues. The CPU extraction queue number is programmable for every interception option in the switch core.

1.9 Synchronous Ethernet and Precision Time Protocol

The device supports Layer-1 ITU-T G.8261 Synchronous Ethernet and Layer-2 IEEE 1588 Precision Time Protocol for synchronizing network timing throughout a network.

Synchronous Ethernet allows for the transfer of network timing from one reference to all network elements. In the device, each port can recover its ingress clock and output the recovered clock to one of up to two recovered clock output pins. External circuitry can then generate a stable reference clock input used for egress and core logic timing in the device.

The Precision Time Protocol (PTP) allows for the network-wide synchronization of precise time of day. It is also possible to derive network timing. PTP can operate with a one-step clock or a two-step clock. For one-step clocks, a frame’s residence time is calculated and stamped into the frame at departure. For two-step clocks, a frame’s residence time is simply recorded and provided to the CPU for further processing. The CPU can then initiate a follow-up message with the recorded timing.

PTP is supported for a range of encapsulations including PTP over Ethernet/IEEE 802.3 and PTP over UDP over IPv4/IPv6.

1.10 CPU System and Interfaces

The device contains a powerful 500 MHz MIPS 24KEc-compatible microprocessor, a high bandwidth Ethernet Frame DMA engine, and a DDR3/DDR3L controller supporting up to 1 gigabyte (GB) of memory. This complete system-on-chip supports Linux or embedded operating systems, enabling full management of the switch and advanced software applications.

The device supports external CPU register access through the on-chip PCIe 1.x endpoint controller, through specially formatted Ethernet frames on the NPI port (Microchip’s Versatile Register Access Protocol), or through register access interface using SPI protocol. External CPUs can inject or extract Ethernet frames through NPI port, through PCIe DMA access, or through register read/writes (using any register-access interface).

2.0 FUNCTIONAL DESCRIPTIONS

This section provides information about the functional aspects of the VSC7513 Industrial IoT Ethernet switch device, available configurations, operational features, and testing functionality.

2.1 Port Numbering and Mappings

The switch core contains an internal port numbering domain where ports are numbered from 0 through 11. A port connects to a port module, which connects to an interface macro, which connects to I/O pins on the device. The interface macros are SERDES1G, SERDES6G, or copper transceivers.

Some ports are internal to the device and do not connect to port modules or interface macros. For example, internal ports are used for frame injection and extraction to the CPU queues.

The following table shows the default port numbering and how the ports map to port modules and interface macros.

TABLE 2-1: DEFAULT PORT NUMBERING AND PORT MAPPINGS

Port Number	Port Module	Maximum Port Speed	Default Interface Macro
0	DEV[0]	1 Gbps	SERDES1G_0 or CUPHY_0
1	DEV[1]	1 Gbps	SERDES1G_1 or CUPHY_1
2	DEV[2]	1 Gbps	SERDES1G_2 or CUPHY_2
3	DEV[3]	1 Gbps	SERDES1G_3 or CUPHY_3
4	DEV[4]	1 Gbps	SERDES1G_4
5	DEV[5]	1 Gbps	SERDES1G_5
6	DEV[6]	1 Gbps	Disconnected
7	DEV[7]	1 Gbps	Disconnected
10 (NPI)	DEV[10]	2.5 Gbps	SERDES6G_2
11 (CPU)	No port module	No macro	No macro

Ports 0 through 3 are dual-media ports and support either a 1G SerDes interface or a 1G copper transceiver.

Ports 11 is an internal port used for injection and extraction of frames towards a CPU. It does not have an associated port module and interface macro.

The following table shows the fixed mapping from interface macros to the device I/O pins.

TABLE 2-2: INTERFACE MACRO TO I/O PIN MAPPING

SERDES Macros	I/O Pin Names
CUPHY_0 to CUPHY_3	P0_D[3:0]N, P0_D[3:0]P - P3_D[3:0]N, P3_D[3:0]P
SERDES1G_0 to SERDES1G_5	S0_RXN, S0_RXP, S0_TXN, S0_TXP - S5_RXN, S5_RXP, S5_TXN, S5_TXP
SERDES6G_0	S6_RXN, S6_RXP, S6_TXN, S6_TXP
SERDES6G_2	S8_RXN, S8_RXP, S8_TXN, S8_TXP

2.1.1 SUPPORTED SERDES INTERFACES

The device support 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 2-3: SUPPORTED SERDES INTERFACES

Port Interface	Specification	Port Speed	Data Rate	Connector	Medium	Coding	SERDES1G	SERDES6G
100BASE-FX	IEEE 802.3, Clause 24	100M	125 Mbps	SFP	PCB	4B5B	x	x
SGMII	Cisco	1G	1.25 Gbps		PCB	8B10B	x	x
SFP	SFP-MSA	1G	1.25 Gbps	SFP	PCB	8B10B	x	x
1000BASE-KX	IEEE802.3, Clause 70	1G	1.25 Gbps		PCB, backplane	8B10B	x	x
2.5G	Proprietary (aligned to SFP)	2.5G	3.125 Gbps		PCB	8B10B		x
QSGMII	Cisco	5G	5 Gbps		PCB	8B10B		x[1]

[1] QSGMII is supported by SERDES6G_0.

2.1.2 DUAL-MEDIA MODE

Ports 0 through 3 are dual-media ports and support either a 1G copper transceiver or a 1G SERDES interface. In dual-media mode, the PCS can be held active while using the copper transceiver. In Half-Duplex mode, the DEV[0-3]:PORT_MODE:PORT_MISC.HDX_FAST_DIS must be set, when the MAC selects the copper tranceiver and resets when the MAC uses the PCS interface.

2.1.3 QSGMII

The SERDES6G_0 macro supports QSGMII. This is enabled in HSIO::HW_CFG.QSGMII_ENA. When QSGMII is enabled, four ports are associated with the SERDES macro to form a QSGMII port with four 1G ports. The muxing related to QSGMII is listed in the following table.

TABLE 2-4: QSGMII MODE

Port Number	Port Module	SERDES Macros
4	DEV[4]	SERDES6G_0
5	DEV[5]	SERDES6G_0
6	DEV[6]	SERDES6G_0
7	DEV[7]	SERDES6G_0

Note that when QSGMII is enabled, the default port connections are removed.

2.1.4 PCIE MODE

SerDes macro SERDES6G_2 can either be used as a standard Ethernet interface or as a PCIe interface. This is controlled in HSIO::HW_CFG.PCIE_ENA. When PCIe is enabled, port 10 cannot be used in option 0 as the SerDes is occupied by the PCIe.

2.1.5 LOGICAL PORT NUMBERS

The analyzer and the rewriter uses in many places 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.

2.2 Port Modules

All port modules contain a MAC. Dual-media ports and ports connecting to a high-speed I/O SerDes macro also contain a PCS.

2.2.1 MAC

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

The MAC supports the following speeds and duplex modes:

- SERDES1G ports: 10/100/1000 Mbps in full-duplex mode and 10/100 Mbps in half-duplex mode
- SERDES6G ports: 10/100/1000/2500 Mbps in full-duplex mode and 10/100 Mbps in half-duplex mode.

The following table lists the registers associated with configuring the MAC.

TABLE 2-5: MAC CONFIGURATION REGISTERS

Register	Description	Replication
DEV::CLOCK_CFG	Reset and speed configuration	Per port
DEV::MAC_ENA_CFG	Enabling of Rx and Tx data paths	Per port
DEV::MAC_MODE_CFG	Port mode configuration	Per port
DEV::MAC_MAXLEN_CFG	Maximum length configuration	Per port
DEV::MAC_TAGS_CFG	VLAN tag length configuration	Per port
DEV::MAC_ADV_CHK_CFG	Type length configuration	Per port
DEV::MAC_IFG_CFG	Interframe gap configuration	Per port
DEV::MAC_HDX_CFG	Half-duplex configuration	Per port
SYS::MAC_FC_CFG	Flow control configuration	Per port
DEV::MAC_FC_MAC_LOW_CFG	LSB of SMAC used in pause frames	Per port
DEV::MAC_FC_MAC_HIGH_CFG	MSB of SMAC used in pause frames	Per port
DEV::MAC_STICKY	Sticky bit recordings	Per port

2.2.1.1 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
- PCS_RX_RST: Reset of the PCS decoder
- PCS_TX_RST: Reset of the PCS encoder

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.

For more information about resetting a port, see Section 4.2.1, Port Reset Procedure.

2.2.1.2 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.
- Data sampling (GIGA_MODE_ENA). Must be 1 in 1 Gbps and 2.5 Gbps and 0 in 10 Mbps and 100 Mbps.
- 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 Gbps (125 MHz clock)
  Ports 8 and 9: 1 Gbps or 2.5 Gbps (125 MHz or 312.5 MHz clock). The actual clock frequency depends on the SerDes configuration.
  100 Mbps (25 MHz clock)
  10 Mbps (2.5 MHz clock)

2.2.1.3 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).

2.2.1.4 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_AWR_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).

2.2.1.5 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 2.2.1.6, 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, see Section 2.11.7, 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_F-C_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.

2.2.1.6 Priority-Based Flow Control (IEEE 802.1Qbb)

The device supports priority-based flow control on all ports for all QoS classes. The following table lists the specific registers associated with priority-based flow control.

TABLE 2-6: PRIORITY-BASED FLOW CONTROL CONFIGURATION REGISTERS

Register	Description	Replication
SYS::MAC_FC_CFG	Flow control configuration	Per port
DEV::MAC_FC_MAC_LOW_CFG	LSB of SMAC used in pause frames	Per port
DEV::MAC_FC_MAC_HIGH_CFG	MSB of SMAC used in pause frames	Per port
ANA::PFC_CFG	Configuration of Rx priority-based flow control per priority.	Per port
QSYS:SWITCH_PORT_MODE	Configuration of Tx priority-based flow control per priority	Per port
DEV::PORT_MISC.FWD_C-TRL_ENA	Enabling forwarding of priority-based pause frames to analyzer.	Per port
ANA::CPU_FWD_BPDU_CFG	Disable forwarding of priority-based pause frames beyond the analyzer	Per port

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. For more information about the watermark, see Table 2-95. 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_F-C_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.

2.2.1.7 Frame Aging

The following table lists the registers associated with frame aging.

TABLE 2-7: FRAME AGING CONFIGURATION REGISTERS

Register	Description	Replication
SYS::FRM_AGING	Frame aging time	None
REW::PORT_CFG.AGE_DIS	Disable frame aging	Per port

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.

Frame aging can be disabled per port (REW::PORT_CFG.AGE_DIS).

Discarded frames due to frame aging are counted in the c_tx_aged counter.

2.2.1.8 Rx and Tx Time Stamps

The following table lists the registers associated with Rx and Tx time stamping.

TABLE 2-8: RX AND TX TIME STAMPING REGISTER

Register	Description	Replication
DEV:PORT_MODE:TX_PATH_DELAY	I/O delays	Per port
ANA:PORT:PTP_CFG.PTP_BACKPLANE_MODE	Ingress backplane configuration	Per port
DEV:PORT_MODE:RX_PATH_DELAY	I/O delays	Per port
DEV:PORT_MODE:PTP_PREDICT_CFG	I/O delays	Per port

The MAC supports Rx and Tx time stamping where a frame's receive time and transmit time are sampled using a nanoseconds counter. The receive and transmit time stamps are shifted in time so that the resulting time stamps align with the exact reception and transmission of the first byte in the frame. This adjusts for local delays in the receive path and the transmit path. The time stamps are individually adjusted as follows:

- Rx time stamp: Sampling of the MAC's nanoseconds counter plus DEV:PORT_MODE:RX_PATH_DELAY.
- Tx time stamp: Sampling of the MAC's nanoseconds counter plus DEV:PORT_MODE:TX_PATH_DELAY.

The Rx and Tx path delays are signed so negative adjustments can be achieved by setting the most significant bit.

The PTP_PREDICT_CFG register must be configured according to the operating mode of the port.

When the ingress port is operating in backplane mode (ANA:PORT:PTP_CFG.PTP_BACKPLANE_MODE), the Rx time stamp is set to the value of the 4-byte reserved field at offset 16 bytes in the incoming frame's PTP header instead of using the adjusted Rx time stamp from the MAC.

The Rx time stamp follows the frame to the rewriter where it can be used to calculate the frame's residence time or it can be sent to the CPU together with the frame. The Tx time stamp is used by the rewriter to calculate the frame's residence time or it can be written to a time stamp FIFO queue accessible by the CPU. For more information about hardware time stamping, see Section 2.17.4, Configuring I/O Delays.

2.2.2 PCS

This section provides information about the Physical Coding Sublayer (PCS) block, where the auto-negotiation process establishes mode of operation for a link. The PCS supports both SGMII mode and two SerDes modes, 1000BASE-X and 100BASE-FX.

The following table lists the registers associated with PCS.

TABLE 2-9: PCS CONFIGURATION REGISTERS

Registers	Description	Replication
PCS1G_CFG	PCS configuration	Per PCS
PCS1G_MODE_CFG	PCS mode configuration	Per PCS
PCS1G_SD_CFG	Signal detect configuration	Per PCS
PCS1G_ANEG_CFG	Configuration of the PCS auto-negotiation process	Per PCS
PCS1G_ANEG_NP_CFG	Auto-negotiation next page configuration	Per PCS
PCS1G_LB_CFG	Loop-back configuration	Per PCS
PCS1G_ANEG_STATUS	Status signaling of the PCS auto-negotiation process	Per PCS
PCS1G_ANEG_NP_STATUS	Status signaling of the PCS auto-negotiation next page process	Per PCS
PCS1G_LINK_STATUS	Link status	Per PCS
PCS1G_LINK_DOWN_CNT	Link down counter	Per PCS
PCS1G_STICKY	Sticky bit register	Per PCS

The PCS is enabled in PCS1G_CFG.PCS_ENA and supports both SGMII and 1000BASE-X SERDES mode (PCS_MODE_CFG.SGMII_MODE_ENA), as well as 100BASE-FX. For information about enabling 100BASE-FX, see Section 2.2.2.7, 100BASE-FX.

The PCS also supports the IEEE 802.3, Clause 66 unidrectional mode, where the transmission of data is independent of the state of the receive link (PCS_MODE_CFG.UNIDIR_MODE_ENA).

2.2.2.1 Auto-Negotiation

Auto-negotiation is enabled in PCS1G_ANEG_CFG.ANEG_ENA. To restart the auto-negotiation process, PCS1G_ANEG_CFG.ANEG_RESTART_ONE_SHOT must be set.

The advertised word for the auto-negotiation process (base page) is configured in PCS1G_ANEG_CFG.ADV_ABILITY. The next page information is configured in PCS1G_ANEG_NP_CFG.NP_TX.

When the auto-negotiation state machine has exchanged base page abilities, the PCS1G_ANEG_STATUS.PAGE_RX_STICKY is asserted indicating that the link partner's abilities were received (PCS1G_ANEG_STATUS.LP_ADV_ABILITY).

If next page information is exchanged, PAGE_RX_STICKY must be cleared, next page abilities must be written to PCS1G_ANEG_NP_CFG.NP_TX, and PCS1G_ANEG_NP_CFG.NP_LOADED_ONE_SHOT must be set. When the auto-negotiation state machine has exchanged the next page abilities, the PCS1G_ANEG_STATUS.PAGE_RX_STICKY is asserted again, indicating that the link partner's next page abilities were received (PCS1G_ANEG_STATUS.LP_NP_RX). Additional exchanges of next page information are possible using the same procedure.

After the last next page is received, the auto-negotiation state machine enters the IDLE_DETECT state and the PCS1G_ANEG_STATUS.PR bit is set indicating that ability information exchange (base page and possible next pages) is finished and software can now resolve priority. Appropriate actions, such as Rx or Tx reset, or auto-negotiation restart, can then be taken, based on the negotiated abilities. The LINK_OK state is reached one link timer period later.

When the auto-negotiation process reaches the LINK_OK state, PCS1G_ANEG_STATUS.ANEG_COMPLETE is asserted.

2.2.2.2 Link Surveillance

The current link status can be observed through PCS1G_LINK_STATUS.LINK_STATUS. The LINK_STATUS is defined as either the PCS synchronization state or as bit 15 of PCS1G_ANEG_STATUS.LP_ADV_ABILTY, which carries information about the link status of the attached PHY in SGMII mode.

Link down is defined as the auto-negotiation state machine being in neither the AN_DISABLE_LINK_OK state nor the LINK_OK state for one link timer period. If a link down event occurs, PCS1G_STICKY.LINK_DOWN_STICKY is set, and PCS1G_LINK_DOWN_CNT is incremented. In SGMII mode, the link timer period is 1.6 ms; in SerDes mode, the link timer period is 10 ms.

The PCS synchronization state can be observed through PCS1G_LINK_STATUS.SYNC_STATUS. Synchronization is lost when the PCS is not able to recover and decode data received from the attached serial link.

2.2.2.3 Signal Detect

The PCS can be enabled to react to loss of signal through signal detect (PCS1G_SD_CFG.SD_ENA). At loss of signal, the PCS Rx state machine is restarted, and frame reception stops. If signal detect is disabled, no action is taken upon loss of signal. The polarity of signal detect is configurable in PCS1G_SD_CFG.SD_POL.

The source of signal detect is selected in PCS1G_SD_CFG.SD_SEL to either the SerDes PMA or the PMD receiver. If the SerDes PMA is used as source, the SerDes macro provides the signal detect. If the PMD receiver is used as source, signal detect is sampled externally through one of the GPIO pins on the device. For more information about the configuration of the GPIOs and signal detect, see Section 3.8.1, SI Boot Controller.

PCS1G_LINK_STATUS.SIGNAL_DETECT contains the current value of the signal detect input.

2.2.2.4 Tx Loopback

For debug purposes, the Tx data path in the PCS can be looped back into the Rx data path. This feature is enabled through PCS1G_LB_CFG.TBI_HOST_LB_ENA.

2.2.2.5 Test Patterns

The following table lists the registers associated with configuring test patterns.

TABLE 2-10: TEST PATTERN REGISTERS

Registers	Description	Replication
PCS1G_TSTPAT_MODE_CFG	Test pattern configuration	Per PSC
PCS1G_TSTPAT_MODE_STATUS	Test pattern status	Per PCS

PCS1G_TSTPAT_MODE_CFG.JTP_SEL overwrites normal operation of the PCS and enables generation of jitter test patterns for debugging. The jitter test patterns are defined in IEEE 802.3, Annex 36A, and the following patterns are supported.

- High frequency test pattern
- Low frequency test pattern
- Mixed frequency test pattern
- Continuous random test pattern with long frames
- Continuous random test pattern with short frames

PCS1G_TSTPAT_MODE_STATUS register holds information about error and lock conditions while running the jitter test patterns.

2.2.2.6 Low Power Idle

The following table lists the registers associated with low power idle (LPI).

TABLE 2-11: LOW POWER IDLE REGISTERS

Registers	Description	Replication
PCS1G_LPI_CFG	Configuration of the PCS low power idle process	Per PSC
PCS1G_LPI_WAKE_ERROR_CNT	Error counter	Per PCS
PCS1G_LPI_STATUS	Low Power Idle status	Per PCS

The PCS supports Energy Efficient Ethernet (EEE) as defined by IEEE 802.3az. The PCS converts Low Power Idle (LPI) encoding between the MAC and the serial interface transparently. In addition, the PCS provides control signals allowing to stop data transmission in the SerDes macro. During low power idles the serial transmitter in the SerDes macro can be powered down, only interrupted periodically while transmitting refresh information, which allows the receiver to notice that the link is still up but in power down mode.

For more information about powering down the serial transmitter in the SerDes macros, see Section 2.3, SERDES1G or Section 2.4, SERDES6G.

It is not necessary to enable the PCS for EEE, because it is controlled indirectly by the shared queue system. It is possible, however, to manually force the PCS into the low power idle mode through PCS1G_LPI_CFG.TX_ASSERT_LPIDLE. During LPI mode, the PCS constantly encodes low power idle with periodical refreshes. For more information about EEE, see Section 2.11.10, Energy Efficient Ethernet.

The current low power idle state can be observed through PCS1G_LPI_STATUS for both receiver and transmitter:

- RX_LPI_MODE: Set if the receiver is in low power idle mode.
- RX_QUIET: Set if the receiver is in the Quiet state of the low power idle mode. If cleared while RX_LPI_MODE is set, the receiver is in the refresh state of the low power idle mode.

The same is observable for the transmitter through TX_LPI_MODE and TX_QUIET.

If an LPI symbol is received, the RX_LPI_EVENT_STICKY bit is set, and if an LPI symbol is transmitted, the TX_LPI_EVENT_STICKY bit is set. These events are sticky.

The PCS1G_LPI_WAKE_ERROR_CNT wake-up error counter increments when the receiver detects a signal and the PCS is not synchronized. This can happen when the transmitter fails to observe the wake-up time or if the receiver is not able to synchronize in time.

2.2.2.7 100BASE-FX

The following table lists the registers associated with 100BASE-FX configuration.

TABLE 2-12: 100BASE-FX REGISTERS

Registers	Description	Replication
PCS_FX100_CFG	Configuration of the PCS 100BASE-FX mode	Per PSC
PCS_FX100_STATUS	Status of the PCS 100BASE-FX mode	Per PCS

The PCS supports a 100BASE-FX mode in addition to the SGMII and 1000BASE-X SerDes modes. The 100BASE-FX mode uses 4-bit/5-bit coding as specified in IEEE 802.3 Clause 24 for fiber connections. The 100BASE-FX mode is enabled through PCS_FX100_CFG.PCS_ENA, which masks out all PCS1G related registers.

The following options are available:

Far-End Fault facility. In 100BASE-FX, the PCS supports the optional Far-End Fault facility. Both Far-End Fault generation (PCS_FX100_CFG.FEF_GEN_ENA) and Far-End Fault Detection (PCS_FX100_CFG.FEF_CHK_ENA) are supported. An Far-End Fault incident is recorded in PCS_FX100_STATUS.FEF_FOUND.

Signal Detect. 100BASE-FX has a similar signal detect scheme to the SGMII and SerDes modes. For 100BASE-FX, PCS_FX100_CFG.SD_ENA enables signal detect, PCS_FX100_CFG.SD_POL controls the polarity, and PCS_FX100_CFG.SD_SEL selects the input source. The current status of the signal detect input can be observed through PCS_FX100_STATUS.SIGNAL_DETECT. For more information about signal detect, see Section 2.2.2.3, Signal Detect.

Link Surveillance. The PCS synchronization status can be observed through PCS_FX100_STATUS.SYNC_STATUS. When synchronization is lost, the link breaks and PCS_FX100_STATUS.SYNC_LOST_STICKY is set. The PCS continuously tries to recover the link.

Unidirectional mode. 100BASE-FX has a similar unidirectional mode as SGMII and SerDes modes. PCS_FX100_CFG.UNIDIR_MODE_ENA enables unidirectional mode.

2.3 SERDES1G

SERDES1G is a high-speed SerDes interface that operates at 1 Gbps (SGMII/SerDes) and 100 Mbps (100BASE-FX). The 100BASE-FX mode is supported by oversampling.

The following table lists the registers associated with SERDES1G.

TABLE 2-13: SERDES1G REGISTERS

Registers	Description	Replication
SERDES1G_COMMON_CFG	Common configuration	Per SerDes
SERDES1G_DES_CFG	Deserializer configuration	Per SerDes
SERDES1G_IB_CFG	Input buffer configuration	Per SerDes
SERDES1G_SER_CFG	Serializer configuration	Per SerDes

TABLE 2-13: SERDES1G REGISTERS (CONTINUED)

Registers	Description	Replication
SERDES1G_OB_CFG	Output buffer configuration	Per SerDes
SERDES1G_PLL_CFG	PLL configuration	Per SerDes
SERDES1G_MISC_CFG	Miscellaneous configuration	Per SerDes

For increased performance in specific application environments, SERDES1G supports the following:

- Programmable loop-bandwidth and phase regulation of deserializer
- Input buffer signal detect/loss of signal (LOS) options
- Input buffer with equalization
- Programmable output buffer features, including:
- De-emphasis
- Amplitude drive levels
- Slew rate control
- Idle mode
- Synchronous Ethernet support
- Loopbacks for system test

2.3.1 SERDES1G BASIC CONFIGURATION

The SERDES1G is enabled in SERDES1G_COMMON_CFG.ENA_LANE. By default, the SERDES1G is held reset and must be released from reset before the interface is active. This is done through SERDES1G_COMMON_CFG.SYS_RST and SERDES1G_MISC_CFG.LANE_RST.

2.3.1.1 SERDES1G Frequency Configuration

To operate the SERDES1G block at 1.25 GHz (corresponding to 1 Gbps data rate), the internal macro PLL must be configured as follows:

1. Configure SERDES1G_PLL_CFG.PLL_FSM_CTRL_DATA to 200.
2. Set SYS_RST = 0 (active) and PLL_FSM_ENA = 0 (inactive).
3. Set SYS_RST = 1 (deactive) and PLL_FSM_ENA = 1 (active).

2.3.2 SERDES1G LOOPBACK MODES

The SERDES1G interface supports two different loopback modes for testing and debugging data paths: equipment loopback and facility loopback.

2.3.2.1 Equipment Loopback (SERDES1G_COMMON_CFG.ENA_ELOOP)

Data is looped back from serializer output to deserializer input, and the receive clock is recovered. The equipment loopback includes all transmit and receive functions, except for the input and output buffers. The Tx data can still be observed on the output.

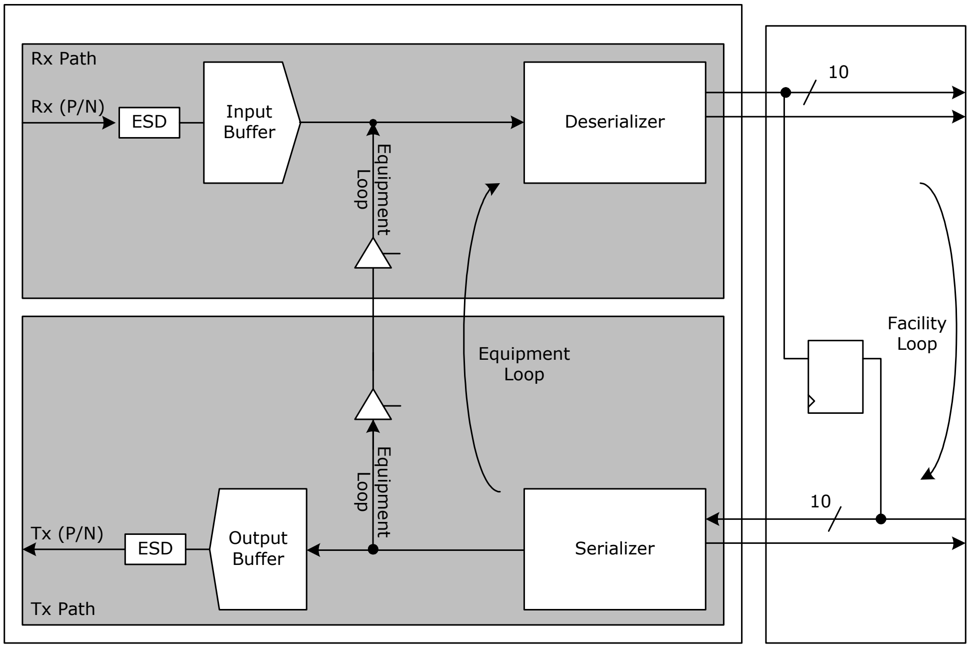
2.3.2.2 Facility Loopback (SERDES1G_COMMON_CFG.ENA_FLOOP)

The clock and parallel data output from deserializer are looped back to the serializer interface. Incoming serial data passes through the input buffer, the CDR, the deserializer, back to the serializer, and finally out through the output buffer.

Only one of the loopbacks can be enabled at the same time.

The following illustration shows the loopback paths.

FIGURE 2-1: SERDES1G LOOPBACK MODES



2.3.3 SYNCHRONOUS ETHERNET

The SERDES1G block can recover the clock from the received data and apply the clock to either a per-SerDes recovered clock output pin (SERDES1G_COMMON_CFG.RECO_SEL_A) or a common recovered clock output pin that may also be driven by another SerDes macros as well (SERDES1G_COMMON_CFG.RECO_SEL_B). Note that only one macro should drive the common recovered clock output pin at the same time.

In addition, it is possible to squelch the recovered clock if the associated PCS cannot detect valid data (SERDES1G_COMMON_CFG.SE_AUTO_SQUELCH_A_ENA and SERDES1G_COMMON_CFG.SE_AUTO_SQUELCH_B_ENA).

For more information about Synchronous Ethernet, see Section 2.16, Layer 1 Timing.

2.3.4 SERDES1G DESERIALIZER CONFIGURATION

The SERDES1G block includes digital control logic that interacts with the analog modules within the block and compensates for the frequency offset between the received data and the internal high-speed reference clock. To gain high jitter performance, the phase regulation is a PI-type regulator, whose proportional (P) and integrative (I) characteristics can be independently configured. The integrative part of the phase regulation loop is configured inSERDES1G_DES_CFG.DES_PHS_CTRL. The limits of the integrator are programmable, allowing different settings for the integrative regulation while guaranteeing that the proportional part still is stronger than the integrative part. Integrative regulation compensates frequency modulation from DC up to cut-off frequency. Frequencies above the cut-off frequency are compensated by the proportional part. The time constant of the integrator is controlled independently of the proportional regulation by SERDES1G_DES_CFG.DES_BW_HYST. The DES_BW_HYST register field is programmable in a range from 3 to 7. The lower the configuration setting, the smaller the time-constant of the integrative regulation. For normal operation, configure DES_BW_HYST to 5.

The cut-off-frequency is calculated to:

fco = 1/(2 × PI × 128 × PLL period × 32 × 2^(DES_BW_HYST + 1 – DES_BW_ANA))

PLL period = 1/(data rate)

The integrative regulator can compensate a static frequency offset within the programmed limits down to a remaining frequency error of below 4 ppm. In steady state, the integrator toggles between two values around the exact value, and the proportional part of the phase regulation takes care of the remaining phase error.

The loop bandwidth for the proportional part of the phase regulation loop is controlled by configuring SERDES1G_DES_CFG.DES_BW_ANA.

The fastest loop bandwidth setting (lowest configuration value) results in a loop bandwidth that is equal to the maximum frequency offset compensation capability. For improved jitter performance, use a setting with sufficient margin to track the expected frequency offset rather than using the maximum frequency offset. For example, if a 100 ppm offset is expected, use a setting that is four times higher than the offset.

The following table provides the limits for the frequency offset compensation. The values are theoretical limits for input signals without jitter, because the actual frequency offset compensation capability is dependent on the toggle rate of the input data and the input jitter. Note that only applicable configuration values are listed.

TABLE 2-14: SERDES1G LOOP BANDWIDTH

DES_BW_ANA	Limits
4	1953 ppm
5	977 ppm
6	488 ppm
7	244 ppm

In the event of an 180° phase jump of the incoming data signal, the sampling stage of the deserializer may become stuck. To prevent this situation, the SERDES1G provides a 180° deadlock protection mechanism (SERDES1G_DES_CFG.DES_MBTR_CTRL). If this mechanism is enabled, a small frequency offset is applied to the phase regulation loop. The offset is sufficient to move the sampling point out of the 180° deadlock region, while at the same time, small enough to allow the regulation loop to compensate when the sample point is within the data eye.

2.3.5 SERDES1G SERIALIZER CONFIGURATION

The serializer provides the ability to align the phase of the internal clock and data to a selected source (SERDES1G_SER_CFG.SER_ENALI). The phase align logic is used when SERDES1G operates in the facility loopback mode.

2.3.6 SERDES1G INPUT BUFFER CONFIGURATION

The SERDES1G input buffer supports the following configurable options:

- 100BASE-FX mode support
- Signal detection, threshold configurable
- Configurable equalization including corner frequency configuration for the equalization filter
- DC voltage offset compensation
- Configurable common mode voltage (CMV) termination
- Selectable hysteresis, configurable hysteresis levels

When the SerDes interface operates in 100BASE-FX mode, the input buffer of the SERDES1G macro must also be configured for 100BASE-FX (SERDES1G_IB_CFG.IB_FX100_ENA).

The input buffer provides an option to configure the threshold level of the signal detect circuit to adapt to different input amplitudes. The signal detect circuit can be configured by SERDES1G_IB_CFG.IB_ENA_DETLEV and SERDES1G_IB_CFG.IB_DET_LEV.

The SERDES1G block offers options to compensate for channel loss. Degraded signals can be equalized, and the corner frequency of the equalization filter can be adapted to the channel behavior. The equalization settings are configured by SERDES1G_IB_CFG.IB_EQ_GAIN and SERDES1G_IB_CFG.IB_CORNER_FREQ.

The SERDES1G block can compensate for possible DC-offset, which can distort the received input signal, by enabling SERDES1G_IB_CFG.IB_ENA_OFFSET_COMP during normal reception.

The common-mode voltage (CMV) input termination can be set to either an internal reference voltage or to VDD_A. To allow external DC-coupling of the input buffer to an output buffer, set the CMV input termination to the internal reference voltage, with internal DC-coupling disabled. SERDES1G_IB_CFG.IB_ENA_DC_COUPLING controls internal DC-coupling, and SERDES1G_IB_CFG.IB_ENA_CMV_TERM controls CMV input termination. The following modes are defined by CMV input termination and DC-coupling.

- SGMII compliant mode with external AC coupling (IB_ENA_DC_COUPLING = 0, IB_ENA_CMV_TERM = 1)
- Microchip-mode with external DC-coupling to another Versatile output buffer, which can operate DC-coupled to the input buffer (IB_ENA_DC_COUPLING = 0, IB_ENA_CMV_TERM = 0)
- 100BASE-FX low frequency mode (IB_ENA_DC_COUPLING = 1, IB_ENA_CMV_TERM = 1)

The SERDES1G macro supports input hysteresis, which is required for some standards (SGMII). The hysteresis function is enabled by SERDES1G_IB_CFG.IB_ENA_HYST, and hysteresis levels are defined by SERDES1G_IB_CFG.IB_HYST_LEV.

Note: Hysteresis and DC offset compensation cannot be enabled at the same time. For more information, see IB_ENA_OFFSET_COMP in the SERDES1G_IB_CFG register.

2.3.7 SERDES1G OUTPUT BUFFER CONFIGURATION

The SERDEDS1G output buffer supports the following configurable options.

- Configurable amplitude settings
- Configurable slew rate control
- 3 dB de-emphasis selectable
- Idle mode

The output amplitude of the output buffer is controlled by SERDES1G_OB_CFG.OB_AMP_CTRL. It can be adjusted in 50 mV steps from 0.4V to 1.1V peak-to-peak differential. The output amplitude also depends on the output buffer's supply voltage. For more information about dependencies between the maximum achievable output amplitude and the output buffer's supply voltage, refer to the electrical section for the dependencies between the maximum achievable output amplitude and the output buffer's supply voltage.

Adjust the slew rate adjustment using SERDES1G_OB_CFG.OB_SLP.

The output buffer supports a fixed 3 dB de-emphasis (SERDES1G_SER_CFG.SER_DEEMPH).

The output buffer supports an idle mode (SERDES1G_SER_CFG.SER_IDLE), which results in an differential peak-to-peak output amplitude of less than 30 mV.

2.3.8 SERDES1G CLOCK AND DATA RECOVERY (CDR) IN 100BASE-FX

To enable clock and data recovery when operating the SERDES1G in 100BASE-FX mode, set the following registers.

- SERDES1G_MISC_CFG.DES_100FX_CPMD_ENA = 1
- SERDES1G_IBJ_CFG.IB_FX100_ENA = 1
- SERDES1G_DES_CFG.DES_CPMD_SEL = 2

2.3.9 ENERGY EFFICIENT ETHERNET

The SERDES1G block supports Energy Efficient Ethernet as defined in IEEE 802.3az. To enable the low power modes, SERDES1G_MISC_CFG.TX_LPI_MODE_ENA and SERDES1G_MISC_CFG.RX_LPI_MODE_ENA must be set. At this point, the attached PCS takes full control over the high-speed output and input buffer activity.

2.3.10 SERDES1G DATA INVERSION

The data streams in the transmit and the receive direction can be inverted using SERDES1G_MISC_CFG.TX_DATA_INV_ENA and SERDES1G_MISC_CFG.RX_DATA_INV_ENA. Effectively this allows for swapping the P and N lines of the high-speed serial link.

2.4 SERDES6G

The SERDES6G is a high-speed SerDes interface that operates at 100 Mbps (100BASE-FX), 1 Gbps (SGMII/SerDes), 2.5 Gbps (SGMII), and 5 Gbps (QSGMII). The 100BASE-FX mode is supported by oversampling.

The following table lists the registers associated with SERDES6G.

TABLE 2-15: SERDES6G REGISTERS

Registers	Description	Replication
SERDES6G_COMMON_CFG	Common configuration	Per SerDes
SERDES6G_DES_CFG	Deserializer configuration	Per SerDes
SERDES6G_IB_CFG	Input buffer configuration	Per SerDes
SERDES6G_IB_CFG1	Input buffer configuration	Per SerDes
SERDES6G_SER_CFG	Serializer configuration	Per SerDes
SERDES6G_OB_CFG	Output buffer configuration	Per SerDes

TABLE 2-15: SERDES6G REGISTERS (CONTINUED)

Registers	Description	Replication
SERDES6G_OB_CFG1	Output buffer configuration	Per SerDes
SERDES6G_PLL_CFG	PLL configuration	Per SerDes
SERDES6G_MISC_CFG	Miscellaneous configuration	Per SerDes

For increased performance in specific application environments, SERDES6G supports the following:

- Baud rate support, configurable from 1 Gbps to 2.5 G, for quarter and half rate modes
- Programmable loop bandwidth and phase regulation for the deserializer
- Configurable input buffer feature such as signal detect/loss of signal (LOS) options
- Configurable output buffer features such as programmable de-emphasis, amplitude drive levels and slew rate control
- Synchronous Ethernet support
- Loopbacks for system test

2.4.1 SERDES6G BASIC CONFIGURATION

The SERDES6G is enabled in SERDES6G_COMMON_CFG.ENA_LANE. By default, the SERDES6G is held reset and must be released from reset before the interface is active. This is done through SERDES6G_COMMON_CFG.SYS-_RST and SERDES6G_MISC_CFG.LANE_RST.

2.4.1.1 SERDES6G PLL Frequency Configuration

To operate the SERDES6G block at the correct frequency, configure the internal macro as follows. PLL calibration is enabled through SERDES6G_PLL_CFG.PLL_FSM_ENA.

- Configure SERDES6G_PLL_CFG.PLL_FSM_CTRL_DATA in accordance with data rates listed in the following tables.
- Set SYS_RST = 0 (active) and PLL_FSM_ENA = 0 (inactive).
- Set SYS_RST = 1 (deactive) and PLL_FSM_ENA = 1 (active).

TABLE 2-16: PLL CONFIGURATION

Mode	SERDES6G_PLL_CFG.PLL_FSM_CTRL_DATA
SGMII/SerDes, 1 Gbps data	60
SGMII, 2.5 Gbps data	48

2.4.1.2 SERDES6G Frequency Configuration

The following table lists the range of data rates that are supported by the SERDES6G block.

TABLE 2-17: SERDES6G FREQUENCY CONFIGURATION REGISTERS

Configuration	SGMII/SerDes 1 Gbps	SGMII 2.5 Gbps
SERDES6G_PLL_CFG.PLL_ROT_FRQ	0	1
SERDES6G_PLL_CFG.PLL_ROT_DIR	1	0
SERDES6G_PLL_CFG.PLL_ENA_ROT	0	1
SERDES6G_COMMON_CFG.QRATE	1	0
SERDES6G_COMMON_CFG.HRATE	0	1

2.4.2 SERDES6G LOOPBACK MODES

The SERDES6G interface supports two different loopback modes for testing and debugging data paths: equipment loopback and facility loopback.

2.4.2.1 Equipment Loopback (SERDES6G_COMMON_CFG.ENA_ELOOP)

Data is looped back from serializer output to the deserializer input, and the receive clock is recovered. The equipment loopback includes all transmit and receive functions, except for the input and output buffers. The Tx data can still be observed on the output.

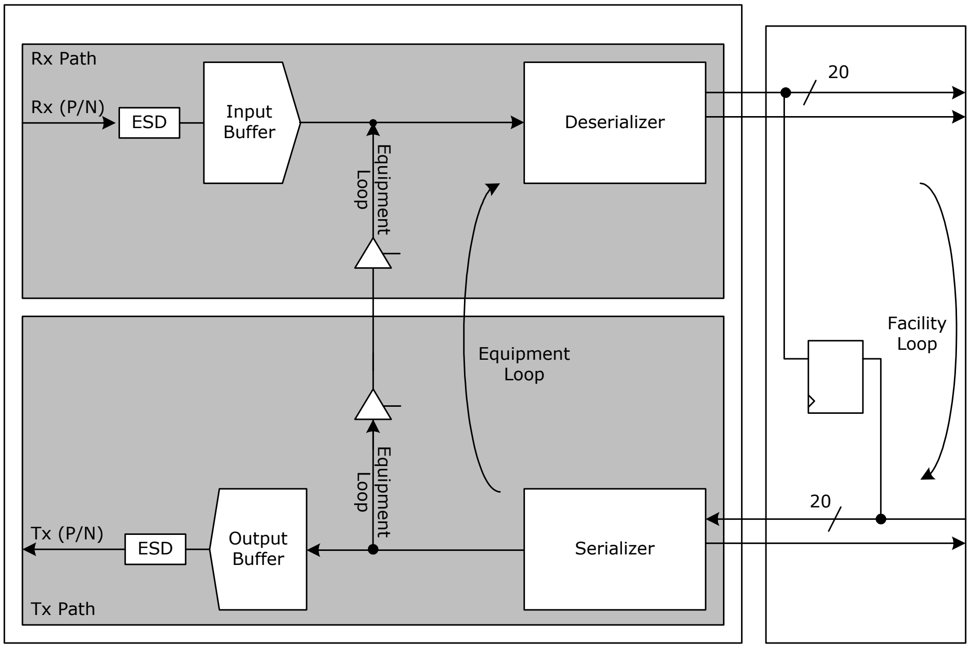
2.4.2.2 Facility Loopback (SERDES6G_COMMON_CFG.ENA_FLOOP)

The clock and parallel data output from deserializer are looped back to the serializer interface. Incoming serial data passes through the input buffer, the CDR, the deserializer, back to the serializer, and finally out through the output buffer.

Only one of the loopbacks can be enabled at the same time.

The following illustration shows the loopback paths for the SERDESG6.

FIGURE 2-2: SERDES6G LOOPBACK MODES



2.4.3 SYNCHRONOUS ETHERNET

The SERDES6G block can recover the clock from the received data and apply the clock to either a per-SerDes recovered clock output pin (SERDES6G_COMMON_CFG.RECO_SEL_A) or a common recovered clock output pin that may be driven by another SerDes macros as well (SERDES6G_COMMON_CFG.RECO_SEL_B). Note that only one macro should drive the common recovered clock output pin at the same time.

In addition, it is possible to squelch the recovered clock if the associated PCS cannot detect valid data (SERDES6G_COMMON_CFG.SE_AUTO_SQUELCH_A_ENA and SERDES6G_COMMON_CFG.SE_AUTO_SQUELCH_B_ENA).

For more information about Synchronous Ethernet, see Section 2.16, Layer 1 Timing.

2.4.4 SERDES6G DESERIALIZER CONFIGURATION

The SERDES6G block includes digital control logic that interacts with the analog modules within the block and compensates for the frequency offset between the received data and the internal high-speed reference clock. To gain high jitter performance, the phase regulation is a PI-type regulator, whose proportional (P) and integrative (I) characteristics can be independently configured.

The integrative part of the phase regulation loop is configured in SERDES6G_DES_CFG.DES_PHS_CTRL. The limits of the integrator are programmable, allowing different settings for the integrative regulation while guaranteeing that the proportional part still is stronger than the integrative part. Integrative regulation compensates frequency modulation from DC up to cut-off frequency. Frequencies greater than the cut-off frequency are compensated by the proportional part.

The DES_BW_HYST register field controls the time constant of the integrator independently of the proportional regulator. The range of DES_BW_HYST is programmable as follows:

- Full rate mode = 3 to 7
- Quarter-rate mode = 1 to 7

The lower the configuration setting the smaller the time-constant of the integrative regulation. For normal operation, configure DES_BW_HYST to 5.

The cut-off-frequency is calculated to:

fco = 1/(2 × PI × 128 × PLL period × 32 × 2^(DES_BW_HYST + 1 – DES_BW_ANA))

PLL period = 1/(n × data rate)

where, `n = 1` (full rate mode) or 4 (quarter-rate mode)

The integrative regulator can compensate a static frequency offset within the programmed limits down to a remaining frequency error of below 4 ppm. In steady state, the integrator toggles between two values around the exact value, and the proportional part of the phase regulation takes care of the remaining phase error. The loop bandwidth for the proportional part of the phase regulation loop is controlled by configuring SERDES6G_DES_CFG.SW.ANA.

The fastest loop bandwidth setting (lowest configuration value) results in a loop bandwidth that is equal to the maximum frequency offset compensation capability. For improved jitter performance, use a setting with sufficient margin to track the expected frequency offset rather than using the maximum frequency offset. For example, if a 100 ppm offset is expected, use a setting that is four times higher than the offset.

The following table provides the limits for the frequency offset compensation. The values are theoretical limits for input signals without jitter, because the actual frequency offset compensation capability is dependent on the toggle rate of the input data and the input jitter. Note that only applicable configuration values are listed. HRATE and QRATE are the configuration settings of SERDES6G_COMMON_CFG.HRATE and SERDES6G_COMMON_CFG.QRATE.

TABLE 2-18: SERDES6G LOOP BANDWIDTH

DES_BW_ANA	Limits when HRATE = 1 QRATE = 0	Limits when HRATE = 0 QRATE = 1
2		1953 ppm
3	1953 ppm	977 ppm
4	977 ppm	488 ppm
5	488 ppm	244 ppm
6	244 ppm	122 ppm
7	122 ppm	61 ppm

In the event of an 180° phase jump of the incoming data signal, the sampling stage of the deserializer may become stuck. To prevent this situation, the SERDES6G provides a 180° deadlock protection mechanism (SERDES6G_DES_CFG.DES_MBTR_CTRL). If this mechanism is enabled, a small frequency offset is applied to the phase regulation loop. The offset is sufficient to move the sampling point out of the 180° deadlock region, while at the same time, small enough to allow the regulation loop to compensate when the sample point is within the data eye.

2.4.5 SERDES6G SERIALIZER CONFIGURATION

The serializer provides the ability to align the phase of the internal clock and data to a selected source (SERDES6G_SER_CFG.SER_ENALI). The phase align logic is used when SERDES6G operates in the facility loopback mode.

2.4.6 SERDES6G INPUT BUFFER CONFIGURATION

The SERDES6G input buffer supports configurable options for:

- Automatic input voltage offset compensation
- Loss of signal detection

The input buffer is typically AC-coupled. As a result, the common-mode termination is switched off (SERDES6G_IB_CFG1.IB_CTERM_ENA). In order to support type-2 loads (DC-coupling at 1.0V termination voltage) according to the OIF CEI specifications, common-mode termination must be enabled.

The sensitivity of the level detect circuit can be adapted to the input signal's characteristics (amplitude and noise). The threshold value for the level detect circuit is set in SERDES6G_IB_CFG.IB_VBCOM. The default value is suitable for normal operation.

When the SerDes interface operates in 100BASE-FX mode, the input buffer of the SERDES6G macro must also be configured for 100BASE-FX (SERDES6G_IB_CFG.IB_FX100_ENA).

During test or reception of low data rate signals (for example, 100BASE-FX), the DC-offset compensation must be disabled. For all other modes, the DC-offset compensation must be enabled for optimized performance. DC-offset compensation is controlled by SERDES6G_IB_CFG1.IB_ENA_OFFSAC and SERDES6G_IB_CFG1.IB_ENA_OFFSDC.

2.4.7 SERDES6G OUTPUT BUFFER CONFIGURATION

The SERDEDS6G output buffer supports the following configurable options.

- Amplitude control
- De-emphasis and output polarity inversion
- Slew rate control
- Skew adjustment
- Idle mode

The maximum output amplitude of the output buffer depends on the output buffer's supply voltage. For interface standards requiring higher output amplitudes (backplane application or interface to optical modules, for example), the output buffer can be supplied from a 1.2V instead of a 1.0V supply. By default, the output buffer is configured for 1.2V mode, because enabling the 1.0V mode when supplied from 1.2V must be avoided. The supply mode is configured by SERDES6G_OB_CFG.OB_ENA1V_MODE.

The output buffer supports a four-tap pre-emphasis realized by one pre-cursor, the center tap, and two post cursors. The pre-cursor coefficient, C0, is configured by SERDES6G_SER_CFG.OB_PREC. C0 is a 5-bit value, with the most significant bit defining the polarity. The lower 4-bit value is hereby defined as B0. The first post-cursor coefficient, C2, is configured by SERDES6G_OB_CFG.OB_POST0. C2 is a 6-bit value, with the most significant bit defining the polarity. The lower 5-bit value is hereby defined as B2. The second post-cursor coefficient, C3, is configured by SERDES6G_SER_CFG.OB_POST1. C3 is 5-bit value, with the most significant bit defining the polarity. The lower 4-bit value is hereby defined as B3. The center-tap coefficient, C1, is a 6-bit value. Its polarity can be programmed by SERDES6G_OB_CFG.OB_POL, which is hereby defined as p1. For normal operation SERDES6G_OB_CFG.OB_POL must be set to 1. The value of the 6 bits forming C1 is calculated by the following equation.

Equation 1: C1: $(64 - (B0 + B2 + B3)) \times p1$

The output amplitude is programmed by SERDES6G_OB_CFG1.OB_LEV, which is a 6-bit value. This value is internally increased by 64 and defines the amplitude coefficient K. The range of K is therefore 64 to 127. The differential peak-peak output amplitude is given by 8.75 mV × K. The maximum peak-peak output amplitude depends on the data stream and can be calculated to:

Equation 2: $H(Z) = 4.375 \text{ mVpp} \times K \times (C0 \times z1 + C1 \times z0 + C2 \times z{-1} + C3 \times z{-2})/64$

with z^n denoting the current bits of the data pattern defining the amplitude of *Z*. The output amplitude also depends on the output buffer's supply voltage.

The configuration bits are summarized in the following table.

TABLE 2-19: DE-EMPHASIS AND AMPLITUDE CONFIGURATION

Configuration	Value	Description
OB_PREC	Signed 5-bit value	Pre-cursor setting *C0*. Range is –15 to 15.
OB_POST0	Signed 6-bit value	First post-cursor setting *C2*. Range is –31 to 31.
OB_POST1	Signed 5-bit value	Second post-cursor setting *C3*. Range is –15 to 15.
OB_LEV	Unsigned 6-bit value	Amplitude coefficient, K = OB_LEV + 64. Range is 0 to 63.
OB_POL	0 1	Non-inverting mode. Inverting mode.

The output buffer provides the following additional options to configure its behavior.

- Idle mode: Enabling idle mode (SERDES6G_OB_CFG.OB_IDLE) results in a remaining voltage of less than 30 mV at the buffers differential outputs.

- Slew Rate: Slew rate can be controlled by two configuration settings. SERDES6G_OB_CFG.OB_SR_H provides coarse adjustments, and SERDES6G_OB_CFG.OB_SR provides fine adjustments.
- Skew control: In 1 Gbps SGMII mode, skew adjustment is controlled by SERDES6G_OB_CFG1.OB_ENA_CAS. Skew control is not applicable to other modes.

2.4.8 SERDES6G CLOCK AND DATA RECOVERY (CDR) IN 100BASE-FX

To enable clock and data recovery when operating SERDES6G in 100BASE-FX mode, set the following register fields:

- SERDES1G_MISC_CFG.DES_100FX_CPMD_ENA = 1
- SERDES1G_IBJ_CFG.IB_FX100_ENA = 1
- SERDES1G_DES_CFG.DES_CPMD_SEL = 2

2.4.9 ENERGY EFFICIENT ETHERNET

The SERDES6G block supports Energy Efficient Ethernet as defined in IEEE Draft P802.3az. To enable the low power modes, SERDES6G_MISC_CFG.TX_LPI_MODE_ENA and SERDES6G_MISC_CFG.RX_LPI_MODE_ENA must be set. At this point, the attached PCS takes full control over the high-speed output and input buffer activity.

2.4.10 SERDES6G DATA INVERSION

The data streams in the transmit and the receive direction can be inverted using SERDES6G_MISC_CFG.TX_DATA_INV_ENA and SERDES6G_MISC_CFG.RX_DATA_INV_ENA. This effectively allows for swapping the P and N lines of the high-speed serial link.

2.4.11 SERDES6G SIGNAL DETECTION ENHANCEMENTS

Signal detect information from the SERDES6G macro is normally directly passed to the attached PCS. It is possible to enable a hysteresis such that the signal detect condition must be active or inactive for a certain time before it is signaled to the attached PCS.

The signal detect assertion time (the time signal detect must be active before the information is passed to a PCS) is programmable in SERDES6G_DIG_CFG.SIGDET_AST. The signal detect de-assertion time (the time signal detect must be inactive before the information is passed to a PCS) is programmable in SERDES6G_DIG_CFG.SIGDET_DST.

2.4.12 HIGH-SPEED I/O CONFIGURATION BUS

The high-speed SerDes macros are configured using the high-speed I/O configuration bus (MCB), which is a serial bus connecting the configuration register set with all the SerDes macros. The HSIO::MCB_SERDES1G_ADDR_CFG register is used for SERDES1G macros and HSIO::MCB_SERDES6G_ADDR_CFG register is used for SERDES6G macros. The configuration busses are used for both writing to and reading from the macros.

The SERDES6G macros are programmed as follows:

- Program the configuration registers for the SERDES6G macro. For more information about configuration options, see Section 2.4, SERDES6G.
- Transfer the configuration from the configuration registers to one or more SerDes macros by writing the address of the macro (MCB_SERDES6G_ADDR_CFG.SERDES6G_ADDR) and initiating the write access (MCB_SERDES6G_ADDR_CFG.SERDES6G_WR_ONE_SHOT).
- The SerDes macro address is a mask with one bit per macro so that one or more macros can be programmed at the same time.
- The MCB_SERDES6G_ADDR_CFG.SERDES6G_WR_ONE_SHOT are automatically cleared when the writing is done.

The configuration and status information in the SERDES6G macros can be read as follows:

- Transfer the configuration and status from one or more SerDes macros to the configuration registers by writing the address of the macro (MCB_SERDES6G_ADDR_CFG.SERDES6G_ADDR) and initiating the read access (MCB_SERDES6G_ADDR_CFG.SERDES6G_RD_ONE_SHOT).
- The SerDes macro address is a mask with one bit per macro so that configuration and status information from one or more macros can be read at the same time. When reading from more than one macro, the results from each macro are OR'ed together.
- The MCB_SERDES6G_ADDR_CFG.SERDES6G_RD_ONE_SHOT are automatically cleared when the reading is done.

The SERDES1G macros are programmed similarly to the SERDES6G macros, except that MCB_SERDES1G_ADDR_CFG must be used for register access. For more information about configuration options, see Section 2.3, SERDES1G.

2.5 Copper Transceivers

This section describes the high-level functionality and operation of four built-in copper transceivers. The integration is kept as close to multichip PHY and switch designs as possible. This allows a fast path for software already running in a similar distributed design while still benefiting from the cost savings provided by the integration.

2.5.1 REGISTER ACCESS

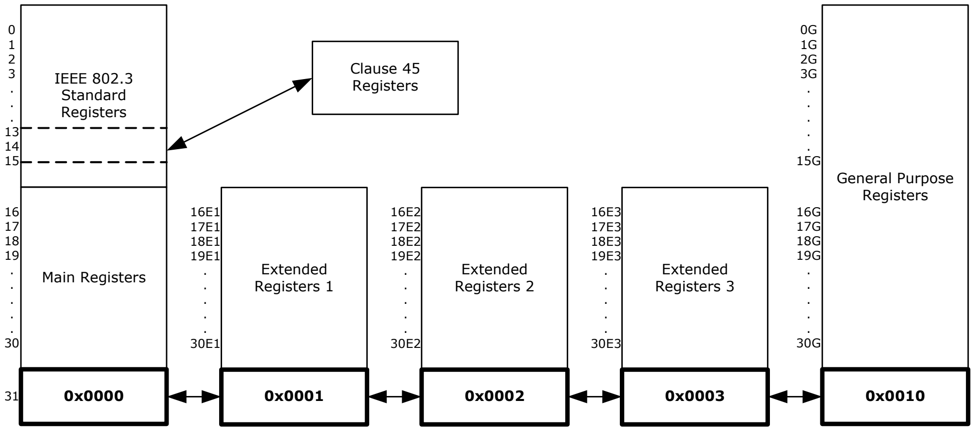
The registers of the integrated transceivers are not placed in the memory map of the switch, but are attached instead to the built-in MII management controller 0 of the device. As a result, PHY registers are accessed indirectly through the switch registers. For more information, see Section 3.8.7, MII Management Controller.

In addition to providing the IEEE 802.3 specified 16 MII Standard Set registers, the PHYs contain an extended set of registers that provide additional functionality. The devices support the following types of registers:

- IEEE Clause 22 device registers with addresses from 0 to 31
- Two pages of extended registers with addresses from 16E1 through 30E1 and 16E2 through 30E2
- General-purpose registers with addresses from 0G to 30G
- IEEE Clause 45 device registers accessible through the Clause 22 registers 13 and 14 to support IEEE 802.3az Energy Efficient Ethernet registers

The memory mapping is controlled through PHY_MEMORY_PAGE_ACCESS::PAGE_ACCESS_CFG. The following illustration shows the relationship between the device registers and their address spaces.

FIGURE 2-3: REGISTER SPACE LAYOUT



2.5.1.1 Broadcast Write

The PHYs can be configured to accept MII PHY register write operations, regardless of the destination address of these writes. This is enabled in PHY_CTRL_STAT_EXT::BROADCAST_WRITE_ENA. This enabling allows similar configurations to be sent quickly to multiple PHYs without having to do repeated MII PHY write operations. This feature applies only to writes; MII PHY register read operations are still interpreted with “correct” address.

2.5.1.2 Register Reset

The PHY can be reset through software, enabled in PHY_CTRL::SOFTWARE_RESET_ENA. Enabling this field initiates a software reset of the PHY. Fields that are not described as sticky are returned to their default values. Fields that are described as sticky are only returned to defaults if sticky-reset is disabled through PHY_CTRL_STAT_EXT::STICKY_RESET_ENA. Otherwise, they retain their values from prior to the software reset. A hardware reset always brings all PHY registers back to their default values.

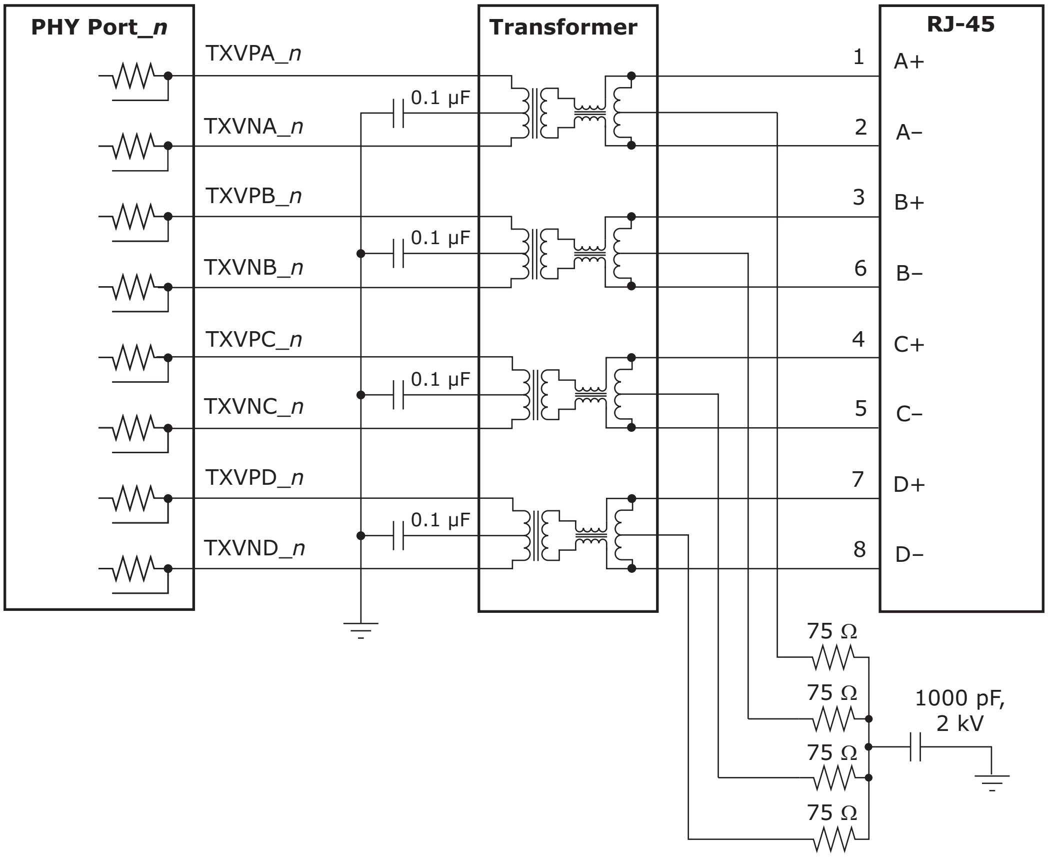
2.5.2 CAT5 TWISTED PAIR MEDIA INTERFACE

The twisted pair interfaces are compliant with IEEE 802.3-2008 and IEEE 802.3az for Energy Efficient Ethernet.

2.5.2.1 Voltage-Mode Line Driver

Unlike many other gigabit PHYs, this PHY uses a patented voltage-mode line driver that allows it to fully integrate the series termination resistors (required to connect the PHY's Cat5 interface to an external 1:1 transformer). Also, the interface does not require placement of an external voltage on the center tap of the magnetic. The following illustration shows the connections.

FIGURE 2-4: CAT5 MEDIA INTERFACE



2.5.2.2 Cat5 Auto-Negotiation and Parallel Detection

The integrated transceivers support twisted pair auto-negotiation as defined by clause 28 of the IEEE 802.3-2008. The auto-negotiation process evaluates the advertised capabilities of the local PHY and its link partner to determine the best possible operating mode. In particular, auto-negotiation can determine speed, duplex configuration, and master or slave operating modes for 1000BASE-TX. Auto-negotiation also allows the device to communicate with the link partner (through the optional "next pages") to set attributes that may not otherwise be defined by the IEEE standard.

If the Cat5 link partner does not support auto-negotiation, the device automatically use parallel detection to select the appropriate link speed.

Auto-negotiation can be disabled by clearing PHY_CTRL.AUTONEG_ENA. If auto-negotiation is disabled, the state of the SPEED_SEL_MSB_CFG, SPEED_SEL_LSB_CFG, and DUPLEX_MODE_CFG fields in the PHY_CTRL register determine the device's operating speed and duplex mode. Note that while 10BASE-T and 100BASE-T do not require auto-negotiation, 1000BASE-T does require it (defined by clause 40).

2.5.2.3 1000BASE-T Forced Mode Support

The integrated transceivers provides support for a 1000BASE-T forced test mode. In this mode, the PHY can be forced into 1000BASE-T mode and does not require manual setting of master/slave at the two ends of the link. This mode is only for test purposes. Do not use in normal operation. To configure a PHY in this mode, set PHY_EEE_CTRL.FORCE_1000BT_ENA = 1, with PHY_CTRL.SPEED_SEL_LSB_CFG = 1 and PHY_CTRL.SPEED_SEL_LSB_CFG = 0.

2.5.2.4 Automatic Crossover and Polarity Detection

For trouble-free configuration and management of Ethernet links, the integrated transceivers include a robust automatic crossover detection feature for all three speeds on the twisted-pair interface (10BASE-T, 100BASE-T, and 1000BASE T). Known as HP Auto-MDIX, the function is fully compliant with clause 40 of the IEEE 802.3-2002.

Additionally, the device detects and corrects polarity errors on all MDI pairs—a useful capability that exceeds the requirements of the standard.

Both HP Auto-MDIX detection and polarity correction are enabled in the device by default. Default settings can be changed using the POL_INV_DIS and PAIR_SWAP_DIS fields in the PHY_BYPASS_CTRL register. Status bits for each of these functions are located in the PHY_AUX_CTRL_STAT register.

The integrated transceivers can be configured to perform HP Auto-MDIX, even when auto-negotiation is disabled (PHY_CTRL.AUTONEG_ENA = 0) and the link is forced into 10/100 speeds. To enable the HP Auto-MDIX feature, set PHY_BYPASS_CTRL.FORCED_SPEED_AUTO_MDIX_DIS to 0.

The HP Auto-MDIX algorithm successfully detects, corrects, and operates with any of the MDI wiring pair combinations listed in the following table.

TABLE 2-20: SUPPORTED MDI PAIR COMBINATIONS

1, 2	3, 6	4, 5	7, 8	Mode
A	B	C	D	Normal MDI
B	A	D	C	Normal MDI-X
A	B	D	C	Normal MDI with pair swap on C and D pair
B	A	C	D	Normal MDI-X with pair swap on C and D pair

2.5.2.5 Manual MDI/MDI-X Setting

As an alternative to HP Auto-MDIX detection, the PHY can be forced to be MDI or MDI X using PHY_EXT_MODE_CTRL.FORCE_MDI_CROSSOVER_ENA. Setting this field to 10 forces MDI, and setting 11 forces MDI-X. Leaving the bits 00 enables the MDI/MDI-X setting to be based on FORCED_SPEED_AUTO_MDIX_DIS and PAIR_SWAP_DIS in the register PHY_BYPASS_CTRL.

2.5.2.6 Link Speed Downshift

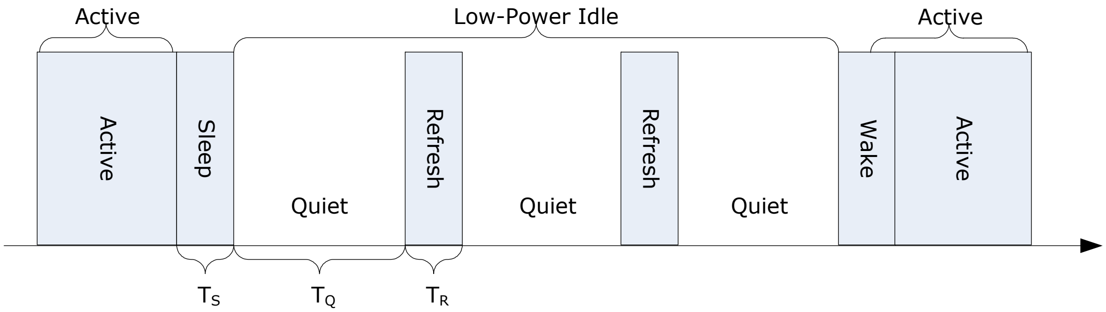
For operation in cabling environments that are incompatible with 1000BASE-T, the device provides an automatic link speed "downshift" option. When enabled, the device automatically changes its 1000BASE-T auto-negotiation advertisement to the next slower speed after a set number of failed attempts at 1000BASE-T. No reset is required to exit this state if a subsequent link partner with 1000BASE-T support is connected. This is useful in setting up in networks using older cable installations that may include only pairs A and B and not pairs C and D.

Link speed downshifting is configured and monitored using SPEED_DOWNSHIFT_STAT, SPEED_DOWNSHIFT_CFG, and SPEED_DOWNSHIFT_ENA in the register PHY_CTRL_EXT3.

2.5.2.7 Energy Efficient Ethernet

The integrated transceivers support IEEE 802.3az Energy Efficient Ethernet (EEE). This standard provides a method for reducing power consumption on an Ethernet link during times of low use.

FIGURE 2-5: LOW POWER IDLE OPERATION



Using LPI, the usage model for the link is to transmit data as fast as possible and then return to a low power idle state. Energy is saved on the link by cycling between active and low power idle states. Power is reduced during LPI by turning off unused circuits and, using this method, energy use scales with bandwidth utilization.

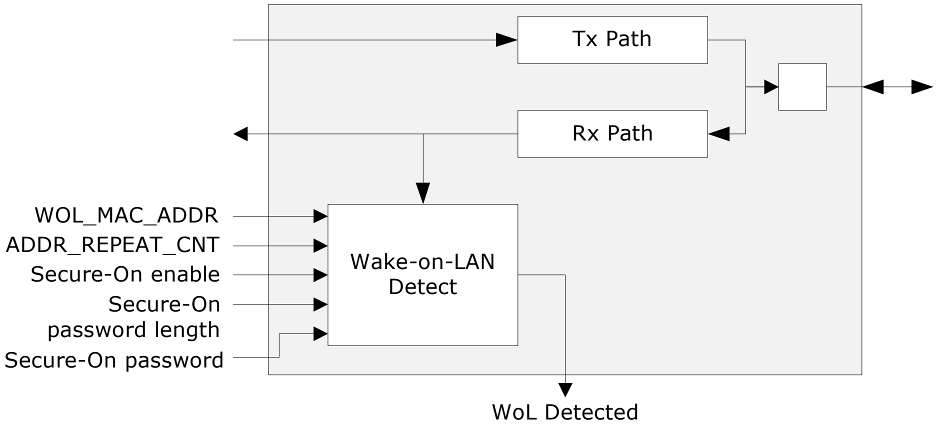
The transceivers use LPI to optimize power dissipation in 100BASE-TX and 1000BASE-T operation. In addition, IEEE 802.3az defines a 10BASE-Te mode that reduces transmit signal amplitude from 5V to approximately 3.3V, peak-to-peak. This mode reduces power consumption in 10 Mbps link speed and can fully interoperate with legacy 10BASE-T compliant PHYs over 100 m Cat5 cable or better.

To configure the transceivers in 10BASE-Te mode, set PHY_EEE_CTRL.EEE_LPI_RX_100BTX_DIS to 1 for each port. Additional Energy Efficient Ethernet features are controlled through Clause 45 registers as defined in Clause 45 registers to Support Energy Efficient Ethernet

2.5.3 WAKE-ON-LAN AND SECUREON

The device supports Wake-on-LAN, an Ethernet networking standard to awaken hosts by using a “magic packet” that is decoded to ascertain the source, and then assert an interrupt pin or an LED. The device also supports SecureOn to secure Wake-on-LAN against unauthorized access. The following illustration shows an overview of the Wake-on-LAN functionality.

FIGURE 2-6: WAKE-ON-LAN FUNCTIONALITY



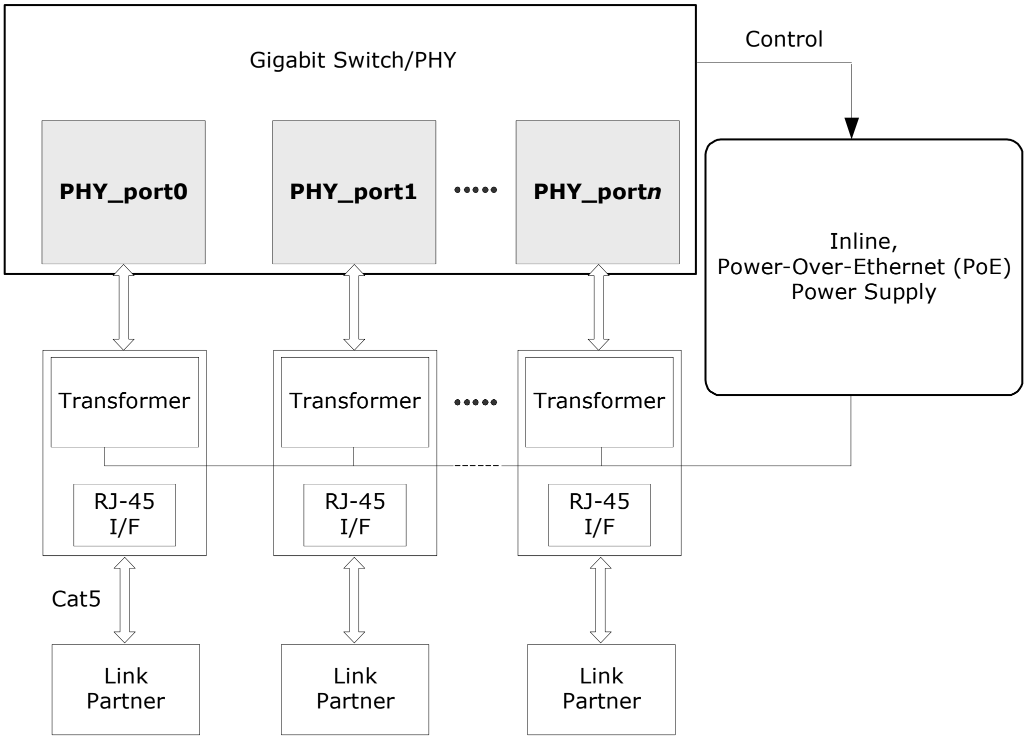
Wake-on-LAN detection is available in 10BASE-T, 100BASE-TX, and 1000BASE-T modes., It is enabled by setting the interrupt mask register PHY_INT_MASK. WOL_INT_ENA and its status is read in the interrupt status register PHY_INT_STAT. WOL_INT_PEND. Wake-on-LAN and SecureOn are configured for each port using the PHY_WOL_MDINT_CTRL register. The MAC address for each port is saved in its local register space (PHY_WOL_MAC_ADDRx).

2.5.4 ETHERNET INLINE POWERED DEVICES

The integrated transceivers can detect legacy inline powered devices in Ethernet network applications. The inline powered detection capability can be part of a system that allows for IP-phone and other devices, such as wireless access points, to receive power directly from their Ethernet cable, similar to office digital phones receiving power from a Private Branch Exchange (PBX) office switch over the telephone cabling. This can eliminate the need of an external power supply for an IP-phone. It also enables the inline powered device to remain active during a power outage (assuming the Ethernet switch is connected to an uninterrupted power supply, battery, back-up power generator, or some other uninterruptable power source).

The following illustration shows an example of this type of application.

FIGURE 2-7: INLINE POWERED ETHERNET SWITCH



The following procedure describes the process that an Ethernet switch must perform to process inline power requests made by a link partner (LP); that is, in turn, capable of receiving inline power.

1. Enables the inline powered device detection mode on each transceiver using its serial management interface. Set PHY_CTRL_EXT4.INLINE_POW_DET_ENA to 1.
2. Ensures that the Auto-Negotiation Enable bit (register 0.12) is also set to 1. In the application, the device sends a special Fast Link Pulse (FLP) signal to the LP. Reading PHY_CTRL_EXT4.INLINE_POW_DET_STAT returns 00 during the search for devices that require Power-over-Ethernet (PoE).
3. The transceiver monitors its inputs for the FLP signal looped back by the LP. An LP capable of receiving PoE loops back the FLP pulses when the LP is in a powered down state. This is reported when PHY_CTRL_EXT4.INLINE_POW_DET_STAT reads back 01. If an LP device does not loop back the FLP after a specific time, PHY_CTRL_EXT4.INLINE_POW_DET_STAT automatically resets to 10.
4. If the transceiver reports that the LP needs PoE, the Ethernet switch must enable inline power on this port, externally of the PHY.
5. The PHY automatically disables inline powered device detection if PHY_CTRL_EXT4.INLINE_POW_DET_STAT automatically resets to 10, and then automatically changes to its normal auto-negotiation process. A link is then auto-negotiated and established when the link status bit is set (PHY_STAT.LINK_STAT is set to 1).
6. In the event of a link failure (indicated when PHY_STAT.LINK_STAT reads 0), the inline power must be disabled to the inline powered device external to the PHY. The transceiver disables its normal auto-negotiation process and re-enables its inline powered device detection mode.

2.5.5 IEEE 802.3AF POE SUPPORT

The integrated transceivers are also compatible with switch designs intended for use in systems that supply power to Data Terminal Equipment (DTE) by means of the MDI or twisted pair cable, as described in clause 33 of the IEEE 802.3af.

2.5.6 ACTIPHY™ POWER MANAGEMENT

In addition to the IEEE-specified power-down control bit (PHY_CTRL.POWER_DOWN_ENA), the device also includes an ActiPHY power management mode for each PHY. The ActiPHY mode enables support for power-sensitive applications. It uses a signal detect function that monitors the media interface for the presence of a link to determine when to automatically power-down the PHY. The PHY “wakes up” at a programmable interval and attempts to wake-up the link partner PHY by sending a burst of FLP over copper media.

The ActiPHY power management mode in the integrated transceivers is enabled on a per-port basis during normal operation at any time by setting PHY_AUX_CTRL_STAT.ACTIPHY_ENA to 1.

Three operating states are possible when ActiPHY mode is enabled:

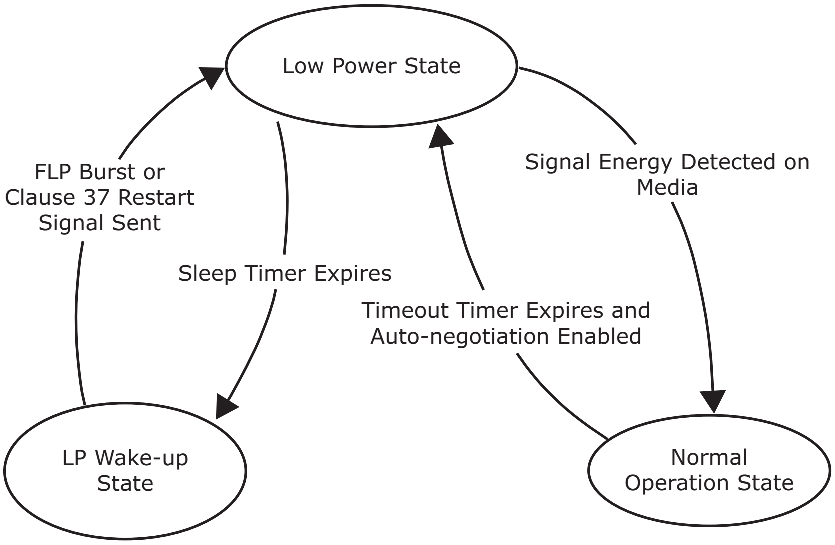
- Low power state
- LP wake-up state
- Normal operating state (link up state)

The PHY switches between the low power state and the LP wake-up state at a programmable rate (the default is two seconds) until signal energy is detected on the media interface pins. When signal energy is detected, the PHY enters the normal operating state. If the PHY is in its normal operating state and the link fails, the PHY returns to the low power state after the expiration of the link status time-out timer. After reset, the PHY enters the low power state.

When auto-negotiation is enabled in the PHY, the ActiPHY state machine operates as described. If auto-negotiation is disabled and the link is forced to use 10BT or 100BTX modes while the PHY is in its low power state, the PHY continues to transition between the low power and LP wake-up states until signal energy is detected on the media pins. At that time, the PHY transitions to the normal operating state and stays in that state even when the link is dropped. If auto-negotiation is disabled while the PHY is in the normal operation state, the PHY stays in that state when the link is dropped and does not transition back to the low power state.

The following illustration shows the relationship between ActiPHY states and timers.

FIGURE 2-8: ACTIPHY STATE DIAGRAM



2.5.6.1 Low Power State

All major digital blocks are powered down in the lower power state.

In this state, the PHY monitors the media interface pins for signal energy. The PHY comes out of low power state and transitions to the normal operating state when signal energy is detected on the media. This happens when the PHY is connected to one of the following:

- Auto-negotiation capable link partner
- Another PHY in enhanced ActiPHY LP wake-up state

In the absence of signal energy on the media pins, the PHY transitions from the low power state to the LP wake-up state periodically based on the programmable sleep timer (PHY_CTRL_EXT3.ACTIPHY_SLEEP_TIMER). The actual sleep time duration is random, from –80 ms to 60 ms, to avoid two linked PHYs in ActiPHY mode entering a lock-up state during operation.

After sending signal energy on the relevant media, the PHY returns to the low power state.

2.5.6.2 Link Partner Wake-up State

In the link partner wake-up state, the PHY attempts to wake up the link partner. Up to three complete FLP bursts are sent on alternating pairs A and B of the Cat5 media for a duration based on the wake-up timer, which is set using register bits 20E1.12:11.

After sending signal energy on the relevant media, the PHY returns to the low power state.

2.5.6.3 Normal Operating State

In normal operation, the PHY establishes a link with a link partner. When the media is unplugged or the link partner is powered down, the PHY waits for the duration of the programmable link status time-out timer, which is set using ACTIPHY_LINK_TIMER_MSB_CFG and ACTIPHY_LINK_TIMER_LSB_CFG in the PHY_AUX_CTRL_STAT register. It then enters the low power state.

2.5.7 TESTING FEATURES

The integrated transceivers include several testing features designed to facilitate performing system-level debugging.

2.5.7.1 Ethernet Packet Generator (EPG)

The Ethernet Packet Generator (EPG) can be used at each of the 10/100/1000BASE-T speed settings for copper Cat5 media to isolate problems between the MAC and the PHY, or between a local PHY and its remote link partner. Enabling the EPG feature effectively disables all MAC interface transmit pins and selects the EPG as the source for all data transmitted onto the twisted pair interface.

Important The EPG is intended for use with laboratory or in-system testing equipment only. Do not use the EPG testing feature when the PHY is connected to a live network.

To use the EPG feature, set PHY_1000BT_EPG2.EPG_ENA to 1.

When PHY_1000BT_EPG2.EPG_RUN_ENA is set to 1, the PHY begins transmitting Ethernet packets based on the settings in the PHY_1000BT_EPG1 and PHY_1000BT_EPG2 registers. These registers set:

- Source and destination addresses for each packet
- Packet size
- Inter-packet gap
- FCS state
- Transmit duration
- Payload pattern

If PHY_1000BT_EPG1.TRANSMIT_DURATION_CFG is set to 0, PHY_1000BT_EPG1.EPG_RUN_ENA is cleared automatically after 30,000,000 packets are transmitted.

2.5.7.2 CRC Counters

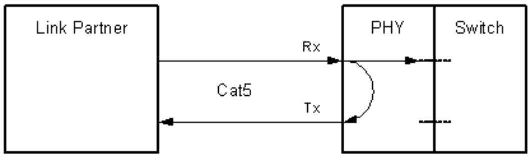
Two separate CRC counters are available in the PHY: a 14-bit good CRC counter available through PHY_CRC_-GOOD_CNT.CRC_GOOD_PKT_CNT and a separate 8-bit bad CRC counter in PHY_C-TRL_EXT4.CRC_1000BT_CNT.

2.5.7.3 Far-End Loopback

The far-end loopback testing feature is enabled by setting PHY_CTRL_EXT1.FAR_END_LOOPBACK_ENA to 1. When enabled, it forces incoming data from a link partner on the current media interface, into the MAC interface of the PHY, to be re-transmitted back to the link partner on the media interface as shown in the following illustration. The incoming data also appears on the receive data pins of the MAC interface. Data present on the transmit data pins of the MAC interface is ignored when using this testing feature.

Note: The far-end loopback is only operational in 1Gbps mode.

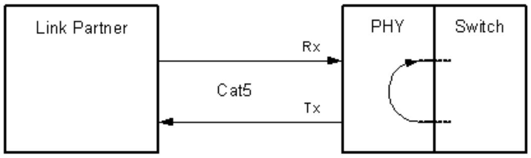
FIGURE 2-9: FAR-END LOOPBACK



2.5.7.4 Near-End Loopback

When the near-end loopback testing feature is enabled (by setting PHY_CTRL.LOOPBACK_ENA to 1), data on the transmit data pins (TXD) is looped back in the PCS block, onto the device receive data pins (RXD), as shown in the following illustration. When using this testing feature, no data is transmitted over the network.

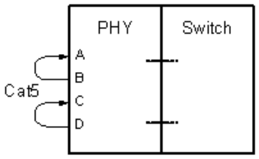
FIGURE 2-10: NEAR-END LOOPBACK



2.5.7.5 Connector Loopback

The connector loopback testing feature allows the twisted pair interface to be looped back externally. When using the connector loopback feature, the PHY must be connected to a loopback connector or a loopback cable. Pair A must be connected to pair B, and pair C to pair D, as shown in the following illustration. The connector loopback feature functions at all available interface speeds.

FIGURE 2-11: CONNECTOR LOOPBACK



When using the connector loopback testing feature, the device auto-negotiation, speed, and duplex configuration is set using device registers 0, 4, and 9. For 1000BASE-T connector loopback, the following additional writes are required, executed in the following steps:

1. Enable the 1000BASE-T connector loopback. Set PHY_CTRL_EXT2.CON_LOOPBACK_1000BT_ENA to 1.
2. Disable pair swap correction. Set PHY_CTRL_EXT2.CON_LOOPBACK_1000BT_ENA to 1.

2.5.8 VERIPHY™ CABLE DIAGNOSTICS

The integrated transceivers include a comprehensive suite of cable diagnostic functions that are available through the onboard processor. These functions enable cable operating conditions and status to be accessed and checked. The VeriPHY suite has the ability to identify the cable length and operating conditions and to isolate common faults that can occur on the Cat5 twisted pair cabling.

For the functional details of the VeriPHY suite and the operating instructions, see the ENT-AN0125, PHY, Integrated PHY-Switch VeriPHY - Cable Diagnostics Feature Application Note.

2.6 Statistics

The following table lists the registers for the statistics module.

TABLE 2-21: COUNTER REGISTERS

Register	Description	Replication
SYS:STAT:CNT	Data register for reading port counters. The current view is specified in SYS::STAT_CFG.STAT_VIEW.	Per counter for specified port
SYS::STAT_CFG.STAT_VIEW	Sets the current counter view.	None
SYS::STAT_CFG.STAT_CLEAR_SHOT	Clears counters per counter group.	None
QSYS::STAT_CNT_CFG.TX_GREEN_CNT_MODE QSYS::STAT_CNT_CFG.TX_YELLOW_CNT_MODE	Controls whether to counts bytes or frames for Tx priority counters.	None
QSYS::STAT_CNT_CFG.DROP_GREEN_CNT_MODE QSYS::STAT_CNT_CFG.DROP_YELLOW_CNT_MODE	Controls whether to counts bytes or frames for drop priority counters.	None
ANA::AGENCTRL.GREEN_COUNT_MODE ANA::AGENCTRL.YELLOW_COUNT_MODE ANA::AGENCTRL.RED_COUNT_MODE	Controls whether to counts bytes or frames for Rx priority counters.	None

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

Each counter is 32 bits wide, which is large enough to ensure a wrap-around time longer than 13 seconds for port statistics on a 2.5G port and longer than 30 seconds for port statistics on a 1G port.

Each switch core port has 44 Rx counters, 18 FIFO drop counters, and 31 Tx counters. The following lists the counters.

2.6.1 PORT STATISTICS

The following table defines the per-port available Rx counters and lists the counter’s address in the common statistics block.

TABLE 2-22: 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	Number of short frames with invalid CRC (<64 bytes).
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.

TABLE 2-22: RECEIVE COUNTERS IN THE STATISTICS BLOCK (CONTINUED)

Type	Short Name	Counter Address	Description
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.
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 yellowby a policer
Rx	c_rx_yellow_prio_1	0x1D	Number of received frames classified to QoS class 1 and marked yellowby a policer
Rx	c_rx_yellow_prio_2	0x1E	Number of received frames classified to QoS class 2 and marked yellowby a policer
Rx	c_rx_yellow_prio_3	0x1F	Number of received frames classified to QoS class 3 and marked yellowby a policer
Rx	c_rx_yellow_prio_4	0x20	Number of received frames classified to QoS class 4 and marked yellowby a policer
Rx	c_rx_yellow_prio_5	0x21	Number of received frames classified to QoS class 5 and marked yellowby a policer
Rx	c_rx_yellow_prio_6	0x22	Number of received frames classified to QoS class 6 and marked yellowby a policer
Rx	c_rx_yellow_prio_7	0x23	Number of received frames classified to QoS class 7 and marked yellowby 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.

TABLE 2-22: RECEIVE COUNTERS IN THE STATISTICS BLOCK (CONTINUED)

Type	Short Name	Counter Address	Description
Rx	c_rx_green_prio_2	0x26	Number of received frames classified to QoS class 2 and marked greenby 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.
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.

The following table defines the per-port available Tx counters and lists the counter address.

TABLE 2-23: TX COUNTERS IN THE STATISTICS BLOCK

Type	Short Name	Counter Address	Description
Tx	c_tx_oct	0x40	Transmitted octets in good and bad frames.
Tx	c_tx_uc	0x41	Number of good unicasts.
Tx	c_tx_mc	0x42	Number of good multicasts.
Tx	c_tx_bc	0x43	Number of good broadcasts.
Tx	c_tx_col	0x44	Number of transmitted frames experiencing a collision. An excessive collided frame gives 16 counts.
Tx	c_txdrop	0x45	Number of frames dropped due to excessive collisions or late collisions.
Tx	c_txpause	0x46	Number of transmitted pause frames.
Tx	c_tx_sz_64	0x47	Number of 64-byte frames in good and bad frames.
Tx	c_tx_sz_65_127	0x48	Number of 65-127-byte frames in good and bad frames.
Tx	c_tx_sz_128_255	0x49	Number of 128-255-byte frames in good and bad frames.
Tx	c_tx_sz_256_511	0x4A	Number of 256-511-byte frames in good and bad frames.
Tx	c_tx_sz_512_1023	0x4B	Number of 512-1023-byte frames in good and bad frames.
Tx	c_tx_sz_1024_1526	0x4C	Number of 1024-1526-byte frames in good and bad frames.
Tx	c_tx_sz_jumbo	0x4D	Number of 1527-MAXLEN.MAX_LENGTH-byte frames in good and bad frames.
Tx	c_tx_yellow_prio_0	0x4E	Number of transmitted frames classified to QoS class 0 with DP level 1.
Tx	c_tx_yellow_prio_1	0x4F	Number of transmitted frames classified to QoS class 1 with DP level 1.
Tx	c_tx_yellow_prio_2	0x50	Number of transmitted frames classified to QoS class 2 with DP level 1.
Tx	c_tx_yellow_prio_3	0x51	Number of transmitted frames classified to QoS class 3 with DP level 1.

TABLE 2-23: TX COUNTERS IN THE STATISTICS BLOCK (CONTINUED)

Type	Short Name	Counter Address	Description
Tx	c_tx_yellow_prio_4	0x52	Number of transmitted frames classified to QoS class 4 with DP level 1.
Tx	c_tx_yellow_prio_5	0x53	Number of transmitted frames classified to QoS class 5 with DP level 1.
Tx	c_tx_yellow_prio_6	0x54	Number of transmitted frames classified to QoS class 6 with DP level 1.
Tx	c_tx_yellow_prio_7	0x55	Number of transmitted frames classified to QoS class 7 with DP level 1.
Tx	c_tx_green_prio_0	0x56	Number of transmitted frames classified to QoS class 0 with DP level 0.
Tx	c_tx_green_prio_1	0x57	Number of transmitted frames classified to QoS class 1 with DP level 0.
Tx	c_tx_green_prio_2	0x58	Number of transmitted frames classified to QoS class 2 with DP level 0.
Tx	c_tx_green_prio_3	0x59	Number of transmitted frames classified to QoS class 3 with DP level 0.
Tx	c_tx_green_prio_4	0x5A	Number of transmitted frames classified to QoS class 4 with DP level 0.
Tx	c_tx_green_prio_5	0x5B	Number of transmitted frames classified to QoS class 5 with DP level 0.
Tx	c_tx_green_prio_6	0x5C	Number of transmitted frames classified to QoS class 6 with DP level 0.
Tx	c_tx_green_prio_7	0x5D	Number of transmitted frames classified to QoS class 7 with DP level 0.
Tx	c_tx_aged	0x5E	Number of frames dropped due to frame aging.

The following table defines the per-port available FIFO drop counters and lists the counter address.

TABLE 2-24: FIFO DROP COUNTERS IN THE STATISTICS BLOCK

Type	Short Name	Counter Address	Description
Drop	c_dr_local	0x80	Number of frames discarded due to no destinations.
Drop	c_dr_tail	0x81	Number of frames discarded due to no more memory in the queue system (tail drop).
Drop	c_dr_yellow_prio_0	0x82	Number of FIFO discarded frames classified to QoS class 0 with DP level 1
Drop	c_dr_yellow_prio_1	0x83	Number of FIFO discarded frames classified to QoS class 1 with DP level 1
Drop	c_dr_yellow_prio_2	0x84	Number of FIFO discarded frames classified to QoS class 2 with DP level 1
Drop	c_dr_yellow_prio_3	0x85	Number of FIFO discarded frames classified to QoS class 3 with DP level 1
Drop	c_dr_yellow_prio_4	0x86	Number of FIFO discarded frames classified to QoS class 4 with DP level 1
Drop	c_dr_yellow_prio_5	0x87	Number of FIFO discarded frames classified to QoS class 5 with DP level 1
Drop	c_dr_yellow_prio_6	0x88	Number of FIFO discarded frames classified to QoS class 6 with DP level 1

TABLE 2-24: FIFO DROP COUNTERS IN THE STATISTICS BLOCK (CONTINUED)

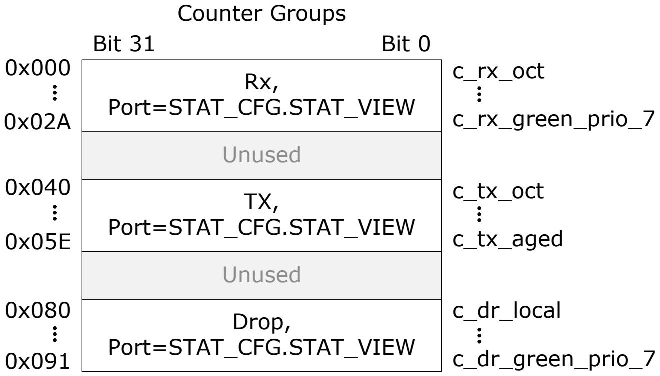
Type	Short Name	Counter Address	Description
Drop	c_dr_yellow_prio_7	0x89	Number of FIFO discarded frames classified to QoS class 7 with DP level 1
Drop	c_dr_green_prio_0	0x8A	Number of FIFO discarded frames classified to QoS class 0 with DP level 0.
Drop	c_dr_green_prio_1	0x8B	Number of FIFO discarded frames classified to QoS class 1 with DP level 0.
Drop	c_dr_green_prio_2	0x8C	Number of FIFO discarded frames classified to QoS class 2 with DP level 0.
Drop	c_dr_green_prio_3	0x8D	Number of FIFO discarded frames classified to QoS class 3 with DP level 0.
Drop	c_dr_green_prio_4	0x8E	Number of FIFO discarded frames classified to QoS class 4 with DP level 0.
Drop	c_dr_green_prio_5	0x8F	Number of FIFO discarded frames classified to QoS class 5 with DP level 0.
Drop	c_dr_green_prio_6	0x90	Number of FIFO discarded frames classified to QoS class 6 with DP level 0.
Drop	c_dr_green_prio_7	0x91	Number of FIFO discarded frames classified to QoS class 7 with DP level 0.

2.6.2 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 are addressable at the same time. The current subset is programmed in STAT_CFG.STAT_VIEW and includes port counters (Rx, Tx, drop) for the port 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 port number to STAT_CFG.STAT_VIEW.

FIGURE 2-12: 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 also read Rx and drop counters on port 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.

For more information about how the counters map to relevant MIBs, see Section 4.2.2, Port Counters.

2.7 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 2.8, VCAP.

2.7.1 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.

The following table lists the registers associated with general data extraction.

TABLE 2-25: GENERAL DATA EXTRACTION REGISTERS

Register	Description	Replication
SYS::PORT_MODE.L3_PARSE_CFG	Enables the use of Layer 3 and 4 protocol information for classification and frame processing.	Per port
SYS::VLAN_ETYPE_CFG	Ethernet Type for S-tags in addition to default value 0x88A8.	None
ANA:PORT.VLAN_CFG.VLAN_INNER_TAG_ENA	Enables using inner VLAN tag for basic classification if available in incoming frame.	Per port
ANA:PORT:S1_VLAN_INNER_TAG_ENA	Enables using inner VLAN tag for IS1 key generation if available in incoming frame.	Per port per IS1 lookup

It is programmable which VLAN tags are recognized. 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 SYS::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.

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 VLAN tags. Frames with more than two VLAN tags are considered non-IP frames.

The actual use of Layer-3 and Layer-4 information for classification, forwarding, and rewriting is enabled in SYS::PORT_MODE.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 Layer4 information
- Analyzer: Flooding and forwarding of IP multicast frames
- Rewriter: Rewriting of IP information

2.7.2 FRAME ACCEPTANCE FILTERING

The following table lists the registers associated with frame acceptance filtering.

TABLE 2-26: FRAME ACCEPTANCE FILTERING REGISTERS

Register	Description	Replication
DEV::PORT_MISC	Configures forwarding of special frames.	Per port
ANA:PORT:DROP_CFG	Configures discarding of illegal frame types.	Per port

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:

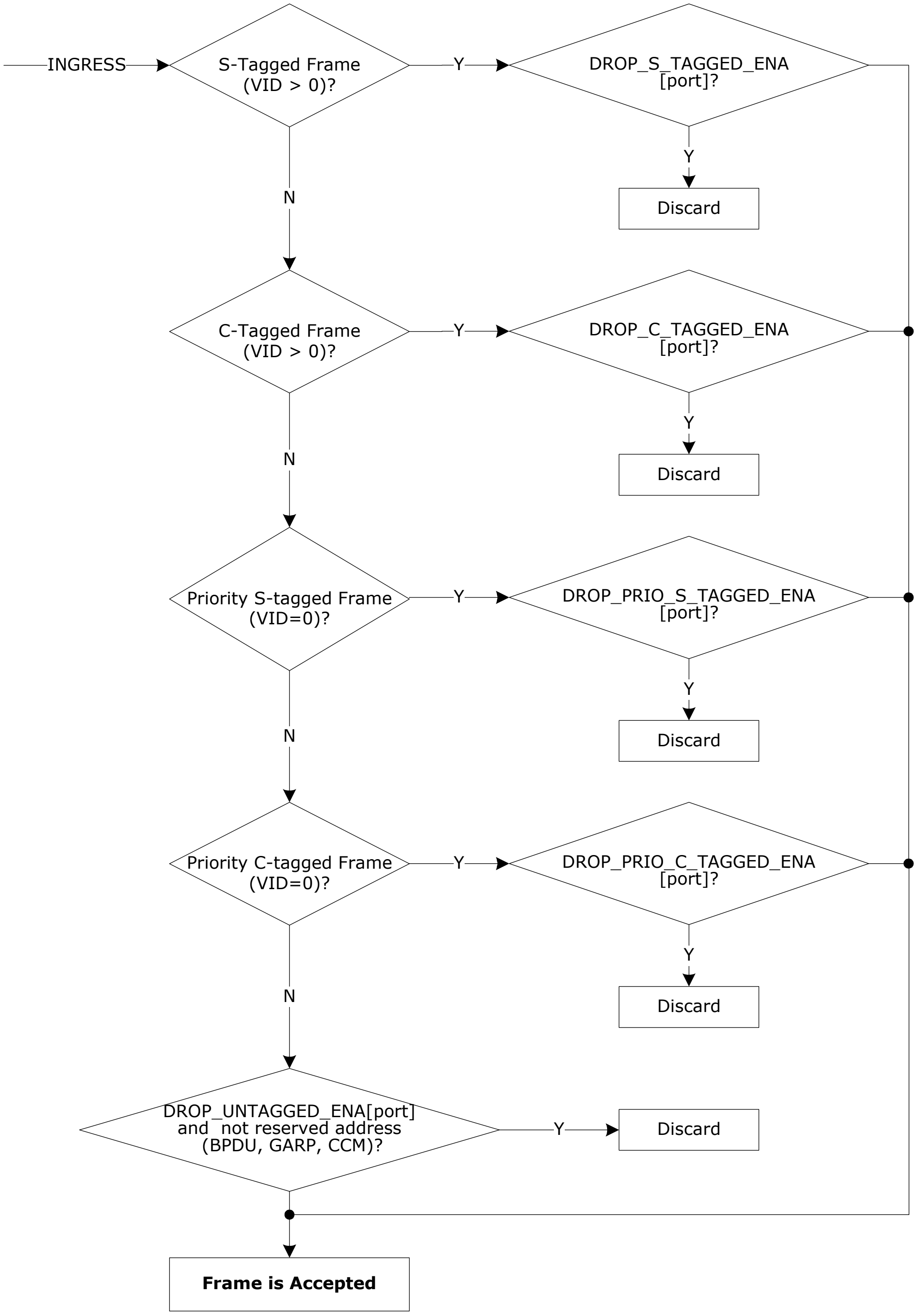
- Frames with a multicast source MAC address
- Frames with a null source or null destination MAC address (address = 0x000000000000)
- Frames with errors signaled by the MAC (for example, an FCS error)
- MAC control frames
- Pause frames after flow control processing in the MAC.
- Untagged frames (excluding 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 (VLAN_CFG.VLAN_INNER_TAG_ENA).

The following illustration shows the flowchart for the VLAN acceptance filter.

FIGURE 2-13: VLAN ACCEPTANCE FILTER


INGRESS
S-Tagged Frame (VID > 0)?
Y
DROP_S_TAGGED_ENA [port]?
Y
Discard
N
C-Tagged Frame (VID > 0)?
Y
DROP_C_TAGGED_ENA [port]?
Y
Discard
N
Priority S-tagged Frame (VID=0)?
Y
DROP_PRIO_S_TAGGED_ENA [port]?
Y
Discard
N
Priority C-tagged Frame (VID=0)?
Y
DROP_PRIO_C_TAGGED_ENA [port]?
Y
Discard
N
DROP_UNTAGGED_ENA[port] and not reserved address (BPDU, GARP, CCM)?
Y
Discard
Frame is Accepted

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.

2.7.3 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 following table lists the registers associated with QoS, DP, and DSCP classification.

TABLE 2-27: QOS, DP, AND DSCP CLASSIFICATION REGISTERS

Register	Description	Replication
ANA.PORT.QOS_CFG	Configuration of the overall classification flow for QoS, DP, and DSCP.	Per port
ANA:PORT:QOS_PCP_DEI_MAP_CFG	Mapping from (DEI, PCP) to (DP, QoS).	Per port per DEI per PCP
ANA::DSCP_CFG	DSCP configuration per DSCP value.	Per DSCP
ANA::DSCP_REWR_CFG	DSCP rewrite values per DP level and QoS class.	Per DP and per QoS

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:

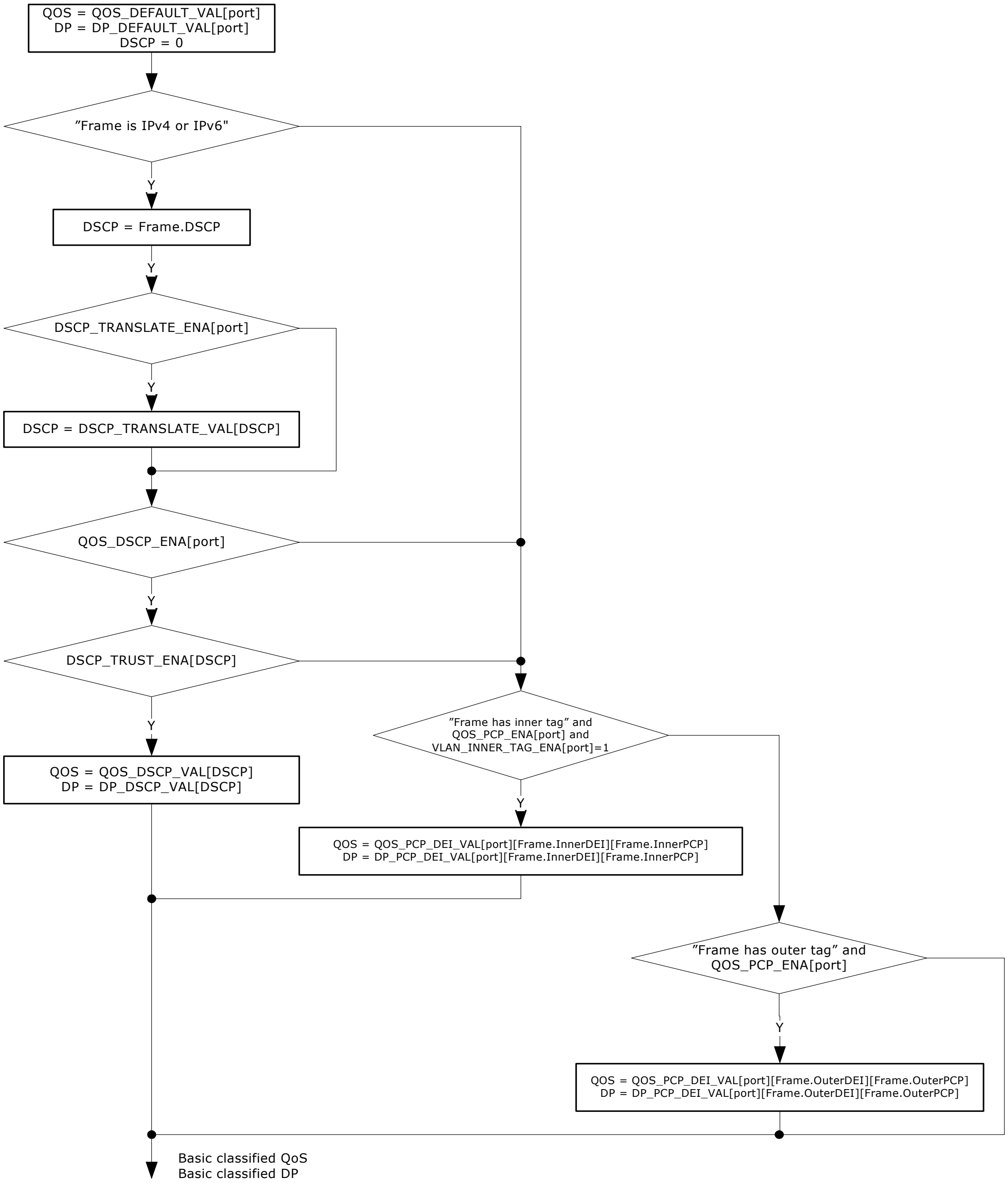
- The frame's QoS class. This class is encoded in a 3-bit field, where 7 is the highest priority QoS class and 0 is the lowest priority QoS class. The QoS class is used by the queue system when enqueuing frames and when evaluating resource consumptions, for policing, statistics, and rewriter actions.
- The frame's 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.

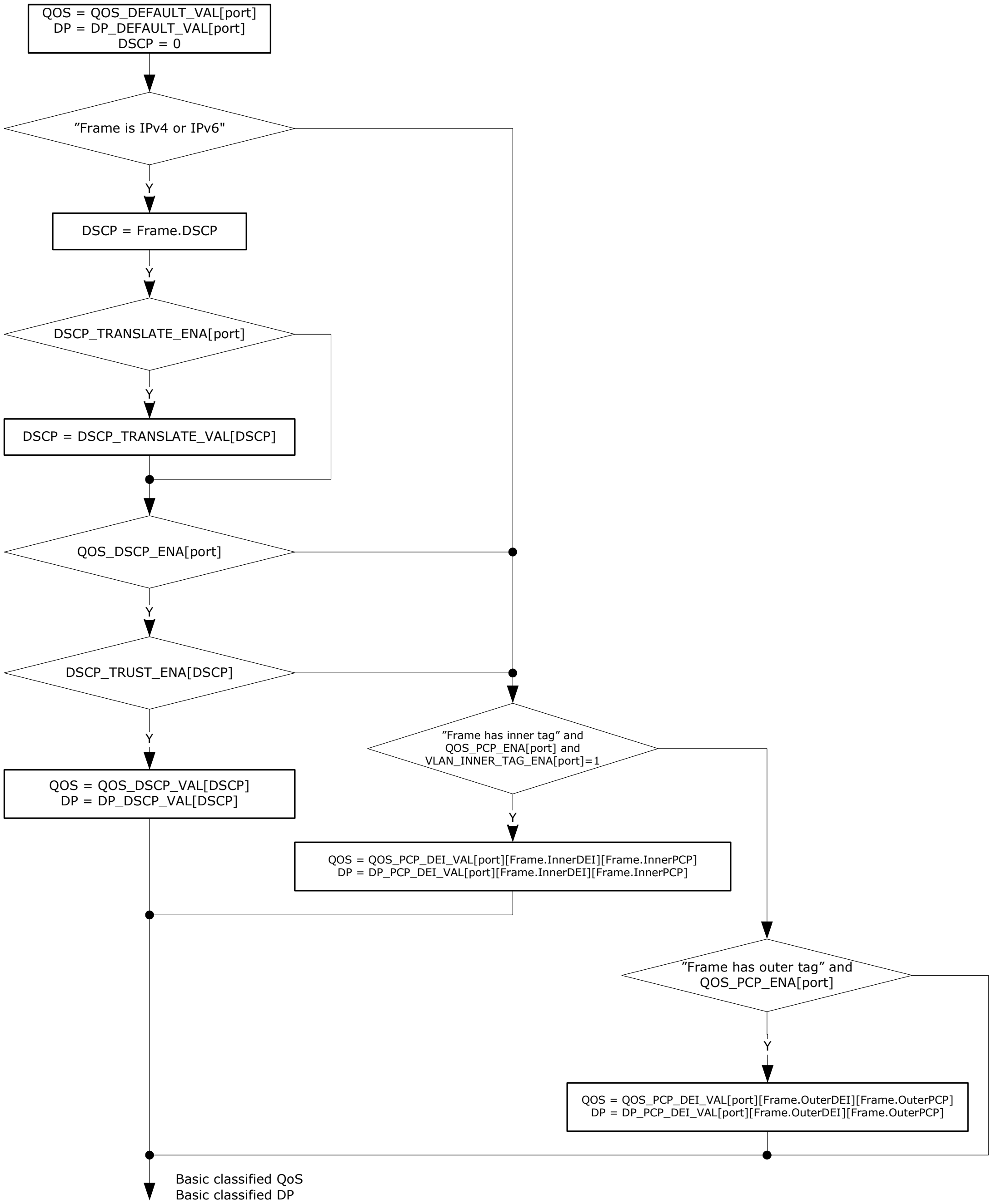
The following illustration shows the flow chart of basic QoS and DP classification.

FIGURE 2-14: QOS AND DP BASIC CLASSIFICATION FLOW


QOS = QOS_DEFAULT_VAL[port]
DP = DP_DEFAULT_VAL[port]
DSCP = 0
”Frame is IPv4 or IPv6"
Y
DSCP = Frame.DSCP
Y
DSCP_TRANSLATE_ENA[port]
Y
DSCP = DSCP_TRANSLATE_VAL[DSCP]
QOS_DSCP_ENA[port]
Y
DSCP_TRUST_ENA[DSCP]
Y
QOS = QOS_DSCP_VAL[DSCP]
DP = DP_DSCP_VAL[DSCP]
”Frame has inner tag” and
QOS_PCP_ENA[port] and
VLAN_INNER_TAG_ENA[port]=1
Y
QOS = QOS_PCP_DEI_VAL[port][Frame.InnerDEI][Frame.InnerPCP]
DP = DP_PCP_DEI_VAL[port][Frame.InnerDEI][Frame.InnerPCP]
”Frame has outer tag” and
QOS_PCP_ENA[port]
Y
QOS = QOS_PCP_DEI_VAL[port][Frame.OuterDEI][Frame.OuterPCP]
DP = DP_PCP_DEI_VAL[port][Frame.OuterDEI][Frame.OuterPCP]
Basic classified QoS
Basic classified DP

The following illustration shows the flow chart for basic DSCP classification.

FIGURE 2-15: BASIC DSCP CLASSIFICATION FLOW CHART


QOS = QOS_DEFAULT_VAL[port]
DP = DP_DEFAULT_VAL[port]
DSCP = 0
”Frame is IPv4 or IPv6"
Y
DSCP = Frame.DSCP
Y
DSCP_TRANSLATE_ENA[port]
Y
DSCP = DSCP_TRANSLATE_VAL[DSCP]
QOS_DSCP_ENA[port]
Y
DSCP_TRUST_ENA[DSCP]
Y
QOS = QOS_DSCP_VAL[DSCP]
DP = DP_DSCP_VAL[DSCP]
”Frame has inner tag” and
QOS_PCP_ENA[port] and
VLAN_INNER_TAG_ENA[port]=1
Y
QOS = QOS_PCP_DEI_VAL[port][Frame.InnerDEI][Frame.InnerPCP]
DP = DP_PCP_DEI_VAL[port][Frame.InnerDEI][Frame.InnerPCP]
”Frame has outer tag” and
QOS_PCP_ENA[port]
Y
QOS = QOS_PCP_DEI_VAL[port][Frame.OuterDEI][Frame.OuterPCP]
DP = DP_PCP_DEI_VAL[port][Frame.OuterDEI][Frame.OuterPCP]
Basic classified QoS
Basic classified DP

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.

2.7.4 VLAN CLASSIFICATION

The following table lists the registers associated with VLAN classification.

TABLE 2-28: VLAN CONFIGURATION REGISTERS

Register	Description	Replication

TABLE 2-28: VLAN CONFIGURATION REGISTERS (CONTINUED)

ANA:PORT:VLAN_CFG	Configures the port's processing of VLAN information in VLAN-tagged and priority-tagged frames. Configures the port-based VLAN.	Per port

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.

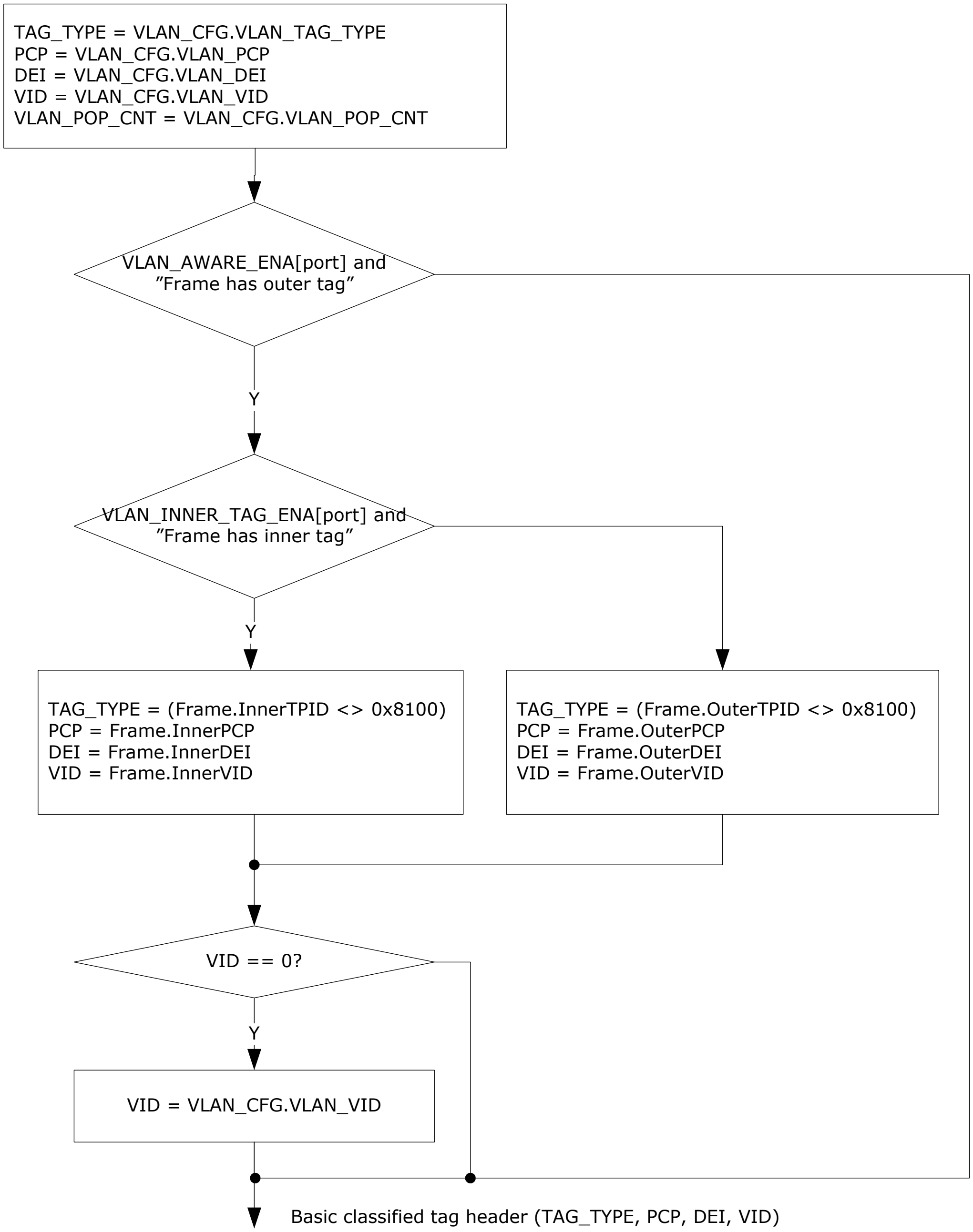
The tag header is either retrieved from a tag in the incoming frame or from a default port-based tag header. The port-based tag header is configured in ANA:PORT:VLAN_CFG.

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

The following illustration shows the flow chart for basic VLAN classification.

FIGURE 2-16: BASIC VLAN CLASSIFICATION FLOW


TAG_TYPE = VLAN_CFG.VLAN_TAG_TYPE
PCP = VLAN_CFG.VLAN_PCP
DEI = VLAN_CFG.VLAN_DEI
VID = VLAN_CFG.VLAN_VID
VLAN_POP_CNT = VLAN_CFG.VLAN_POP_CNT
VLAN_AWARE_ENA[port] and ”Frame has outer tag”
Y
VLAN_INNER_TAG_ENA[port] and ”Frame has inner tag”
Y
TAG_TYPE = (Frame.InnerTPID <> 0x8100)
PCP = Frame.InnerPCP
DEI = Frame.InnerDEI
VID = Frame.InnerVID
TAG_TYPE = (Frame.OuterTPID <> 0x8100)
PCP = Frame.OuterPCP
DEI = Frame.OuterDEI
VID = Frame.OuterVID
VID == 0?
Y
VID = VLAN_CFG.VLAN_VID
Basic classified tag header (TAG_TYPE, PCP, DEI, VID)

The basic classifier can overwrite the basic classified PCP value with the frame’s basic classified QoS class coming from the basic QoS classification. This option is enabled in ANA:PORT:PTP_DLY1_CFG.USE_QOS_AS_PCP_ENA. The basic classified tag header can be overwritten by more intelligent decisions made in the VCAP IS1.

2.7.5 LINK AGGREGATION CODE GENERATION

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

The following table lists the registers associated with aggregation code generation.

TABLE 2-29: AGGREGATION CODE GENERATION REGISTERS

Register	Description	Replication
ANA::AGGR_CFG	Configures use of Layer-2 through Layer-4 flow information for link aggregation code generation.	Common

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. For more information about how the aggregation code is used, see Section 4.3.8, Link Aggregation.

2.7.6 CPU FORWARDING DETERMINATION

The following table lists the registers associated with CPU forwarding in the basic classifier.

TABLE 2-30: CPU FORWARDING DETERMINATION

Register	Description	Replication
CPU_FWD_CFG	Enables CPU forwarding for various frame types.	Per port
CPU_FWD_BPDU_CFG	Enables CPU forwarding per BPDU address.	Per port
CPU_FWD_GARP_CFG	Enables CPU forwarding per GARP address.	Per port
CPU_FWD_CCM_CFG	Enables CPU forwarding per CCM/Link trace address	Per port
CPUQ_CFG and CPUQ_CFG2	CPU extraction queues for various frame types	None
CPUQ_8021_CFG	CPU extraction queues for BPDU, GARP, and CCM addresses.	None
VRAP_CFG	VLAN configuration of VRAP filter.	None
VRAP_HDR_DATA	Data match against VRAP header.	None
VRAP_HDR_MASK	Mask used to don't care bits in the VRAP header.	None

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. For more information, see Section 4.7, CPU Extraction and Injection.

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.

The following table lists the standard frame types, with respect to CPU forwarding, that are recognized by the classifier.

TABLE 2-31: 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. For more information, see Section 2.15, VRAP Engine.

To determine if a frame is a VRAP frame, the VRAP filter in the classifier performs three checks:

- 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.
- EtherType and EPID check. The EtherType must be 0x8880 and the EPID must be 0x0004 (bytes 0 and 1 after the EtherType).
- 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.

2.8 VCAP

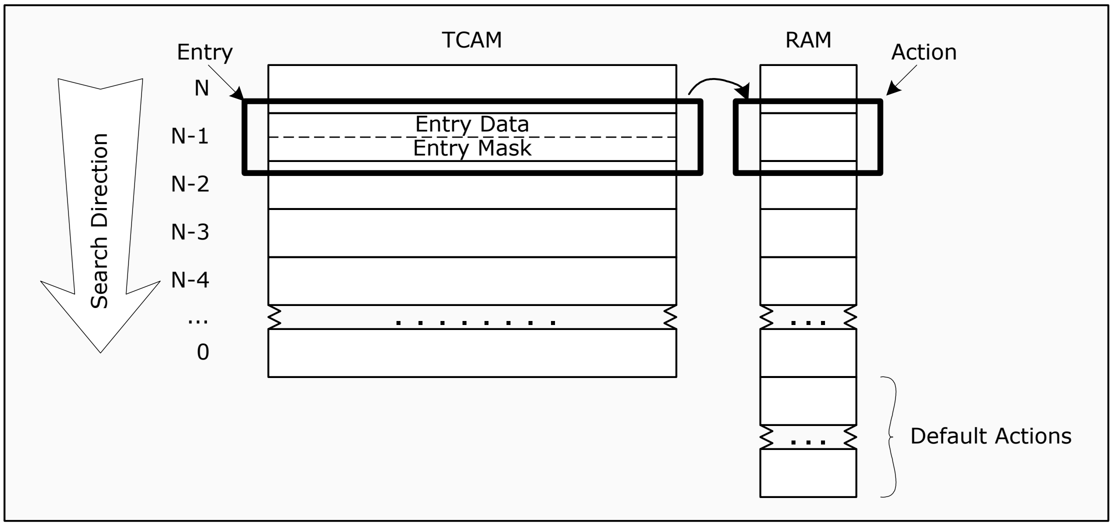
The VCAP is a content-aware packet processor for wire-speed packet inspection for rich implementation of, for example, advanced VLAN and QoS classifications and manipulations, IP source guarding, and security features for wireline and wireless applications.

The following describes three VCAPs implemented in the device: IS1, IS2, and ES0. IS1 and IS2 are generic ingress VCAPs working on the incoming frames while ES0 is an egress VCAP working on all outgoing frames.

When a VCAP is enabled, each frame is examined to determine the frame type (for example IPv4 TCP frame) so that the frame information is extracted according to the frame type. Together with port-specific configuration and classification results from the basic classification, the extracted frame information makes up an entry key, which is passed to a TCAM and matched against entries in the TCAM.

An entry in the TCAM consists of a pattern and a mask, where the mask allows pattern-matching with the use of "don't cares". The first matching entry is then used to select an action. The following illustration provides a functional overview of a general TCAM.

FIGURE 2-17: VCAP FUNCTIONAL OVERVIEW



Each frame results in up to six ingress VCAP lookups and one egress lookup per destination port. The lookups use different keys and the results determine the frame's ingress classification, security handling, and egress VLAN manipulation. The six ingress lookups and the associated VCAPs are:

- Advanced ingress classification, first lookup:
- VCAP: IS1
- Key: IS1
- Entry: IS1 Control Entry
- Advanced ingress classification, second lookup:
- VCAP: IS1
- Key: IS1
- Entry: IS1 Control Entry
- Advanced ingress classification, third lookup:
- VCAP: IS1
- Key: IS1
- Entry: IS1 Control Entry
- IP source guarding check:
- VCAP: IS2
- Key: SMAC_SIP4 (IPv4 frames) or SMAC_SIP6 (IPv6 frames)
- Entry: SMAC_SIP4 Control Entry or SMAC_SIP6 Control Entry
- Security enforcement, first lookup:
- VCAP: IS2
- Key: MAC_ETYPE, MAC_LLC, MAC_SNAP, ARP, IP4_OTHER, IP4_TCP_UDP, IP6_STD, IP6_OTHER, IP6_TCP_UDP, OAM, CUSTOM, depending on frame type
- Entry: Access Control Entry
- Security enforcement, second lookup
- VCAP: IS2

- Key: MAC_ETYPE, MAC_LLC, MAC_SNAP, ARP, IP4_OTHER, IP4_TCP_UDP, IP6_STD, IP6_OTHER, IP6_TCP_UDP, OAM, CUSTOM, depending on frame type
- Entry: Access Control Entry

The egress lookup per destination port and associated VCAP is:

- Egress tagging and frame manipulations
- VCAP: ES0
- Key: ES0
- Entry: Egress Control Entry

The IP source guarding check is only carried out for IP frames.

CPU injected frames are subject to all the above VCAP lookups in IS1, IS2, and ES0.

With respect to IS1 and IS2, each frame is classified to one of seven overall VCAP frame types. The frame type determines the information to extract from the frame and also which VCAP entries to match against. The following table lists which frame types are used and which VCAP entries the frame types are matched against in IS1 and IS2. **Note** The lookup in ES0 is independent of the frame type, and all frames match against all entries in the TCAM.

TABLE 2-32: IS1 AND IS2 VCAP FRAME TYPES

Frame Type	Condition	IS1 Entries	IS2 Entries

IPv6 Frame	The Type/Len field is equal to 0x86DD.
	The IP version is 6.
	Special IPv6 frames:
	•IPv6 TCP frame: Next header is TCP (0x6)
	•IPv6 UDP frame: Next header is UDP (0x11)
	•IPv6 Other frame: Next header is neither TCP nor UDP
		Frame type flags:
		ETYPE_LEN = 1
		IP_SNAP = 1
		IP4 = 0
		TCP_UDP
		TCP
			IP6_TCP_UDP
			IP6_OTHER
			IP6_STD

IPv4 Frame	The Type/Len field is equal to 0x800.
	The IP version is 4.
	Special IPv4 frames:
	•IPv4 TCP frame: IP protocol is TCP (0x6)
	•IPv4 UDP frame: IP protocol is UDP (0x11)
	•IPv4 Other frame: IP protocol is neither TCP nor UDP
		Frame type flags:
		ETYPE_LEN = 1
		IP_SNAP = 1
		IP4 = 1
		TCP_UDP
		TCP
			IP4_TCP_UDP
			IP4_OTHER

(R)ARP Frame	The Type/Len field is equal to 0x0806 (ARP) or 0x8035 (RARP).	Frame type flags:
		ETYPE_LEN = 1
		IP_SNAP = 0
			ARP

OAM Frame	The Type/Len field is equal to 0x8902, 0x8809, or 0x88EE.
	In particular, the following protocol data units use this EtherType:
	•IEEE 802.1ag (CFM)
	•IEEE 802.3 (EFM)
	•ITU-T Y.1731 (ETH-OAM)
	•MEF-16 (E-LMI)
		Frame type flags:
		ETYPE_LEN = 1
		IP_SNAP = 0
			OAM

SNAP Frame	The Type/Len field is less than 0x600.
	The Destination Service Access Point field, DSAP is equal to 0xAA.
	The Source Service Access Point field, SSAP is equal to 0xAA.
	The Control field is equal to 0x3.
		Frame type flags:
		ETYPE_LEN = 0
		IP_SNAP = 1
			MAC_SNAP

LLC Frame	The Type/Len field is less than 0x600
	The LLC header does not indicate a SNAP frame.
		Frame type flags:
		ETYPE_LEN = 0
		IP_SNAP = 0
			MAC_LLC

TABLE 2-32: IS1 AND IS2 VCAP FRAME TYPES (CONTINUED)

Frame Type	Condition	IS1 Entries	IS2 Entries

ETYPE Frame	The Type/Len field is greater than or equal to 0x600.
	The Type field does not indicate any of the previously mentioned frame types, that is, ARP, RARP, IPv4, or IPv6.
		Frame type flags:
		ETYPE_LEN = 1
		IP_SNAP = 0
			MAC_ETYPE

In addition, Precision Time Protocol (PTP) frames are handled by IS2. The following encapsulations of PTP frames are supported:

- PTP over Ethernet:
  ETYPE frame with Type/Len = 0x88F7.
  Matched against MAC_ETYPE entries.
- PTP over UDP over IPv4:
  IPv4 UDP frame with UDP destination port numbers 319 or 320.
  Matched against IP4_TCP_UDP entries.
- PTP over UDP over IPv6
  IPv6 UDP frame with UDP destination port numbers 319 or 320.
  Matched against IP6_TCP_UDP entries.

PTP frames may be untagged, single tagged, or double tagged.

For PTP over Ethernet, the following PTP fields are always extracted:

- TransportSpecific (byte 0)
- MessageType (byte 0)
- VersionPTP (byte 1)
- DomainNumber (byte 4)
- FlagField: flags 1, 2, and 7 (byte 6)

Note Byte 0 is the byte immediately following the EtherType, then byte 1, byte 2, and so on.

For PTP over UDP, the following PTP fields are always extracted:

- MessageType (byte 0)
- VersionPTP (byte 1)
- DomainNumber (byte 4)
- FlagField: flags 1, 2, and 7 (byte 6)

Note Byte 0 is the byte immediately following the UDP header, then byte 1, byte 2, and so on.

2.8.1 PORT CONFIGURATION

This section provides information about special port configurations that control the key generation for the VCAPs.

The following table lists the registers associated with port configuration for VCAP.

TABLE 2-33: PORT MODULE CONFIGURATION OF VCAP

Register	Description	Replication
ANA:PORT:VCAP_CFG	Configuration of the key generation for the VCAPs.	Per port
ANA:PORT:VCAP_S1_KEY_CFG	Configuration of the key generation for the VCAP IS1 per lookup.	Per port
ANA:PORT:VCAP_S2_CFG	Configuration of the key generation for the VCAP IS2.	Per port
REW:PORT:PORT_CFG	Enables VCAP ES0.	Per port

Each port module affects the key generation for VCAPs IS1 and IS2 through the VCAP_CFG, VCAP_S1_KEY_CFG, and VCAP_S2_CFG registers, and the rewriter affects VCAP ES0 through the REW:PORT:PORT_CFG.ES0_ENA register.

2.8.1.1 VCAP IS1 Port Configuration

The following port configurations are available for IS1:

- Enable lookups in IS1 (VCAP_CFG.S1_ENA). If disabled, frames received by the port module are not matched against rules in VCAP IS1.
- Use destination information rather than source information (VCAP_CFG.S1_DMAC_DIP_ENA). By default, the two advanced classification lookups in IS1 use the source MAC address and source IP address from the incoming frame when generating the key. Through S1_DMAC_DIP_ENA, the corresponding destination information, destination MAC address, and destination IP address can be used instead. This can be controlled per lookup so that, for example, the first lookup applies source information, and the second applies destination information.
- Use inner VLAN tag rather than outer VLAN tag (VCAP_CFG.S1_VLAN_INNER_TAG_ENA). By default, the two advanced classification lookups in IS1 use the outer VLAN tag from the incoming frame when generating the key. Through S1_VLAN_INNER_TAG_ENA, the inner tag for double tagged frames can be used. This can be controlled per lookup so that, for example, the first lookup applies the outer tag, and the second lookup applies the inner tag. For single tagged frames, the outer VLAN tag is always used.
- Select which IS1 key to use for IPv6 frames (VCAP_S1_CFG.S1_KEY_IP6_CFG). This can be controlled per lookup in IS1. IPv6 frames can use any of the six supported keys in IS1. For more information about the IS1 keys, see Section 2.8.2, VCAP IS1.
- Select which IS1 key to use for IPv4 frames (VCAP_S1_CFG.S1_KEY_IP4_CFG). This can be controlled per lookup in IS1. IPv4 frames can use S1_NORMAL, S1_7TUPLE, S1_5TUPLE_IP4, or S1_DBL_VID keys.
- Select which IS1 key to use for non-IP frames (VCAP_S1_CFG.S1_KEY_OTHER_CFG). This can be controlled per lookup in IS1. Non-IP frames can use S1_NORMAL, S1_7TUPLE, or S1_DBL_VID keys.

2.8.1.2 VCAP IS2 Port Configuration

The following port configurations are available for IS2:

- Enable lookups in IS2 (VCAP_S2_CFG.S2_ENA). If disabled, frames received by the port module are not matched against rules in VCAP IS2.
- Default PAG value (VCAP_CFG.PAG_VAL). This PAG value is the initial value. Actions out of IS1 can change the PAG value before it is used in the key for IS2.

Each port module can control a hierarchy of which entry types in IS2 to use for different frame types. This is controllable per lookup. For instance, it is controllable whether IPv6 TCP frames are matched against IP6_TCP_UDP entries, IP6_STD entries, IP4_TCPUDP entries, or MAC_ETYPE entries. Note that matching against an entry type controls how the key is generated.

With reference to the VCAP_S2_CFG register, the following table lists the hierarchy for different frame types.

TABLE 2-34: HIERARCHY OF IS2 ENTRY TYPES

Frame Type	Description
IPv6 TCP and UDP frames	Configuration: S2_IP6_CFG. If S2_IP6_CFG is set to 0, IPv6 TCP and UDP frames are matched against IP6_TCP_UDP entries. If S2_IP6_CFG is set to 1, IPv6 TCP and UDP frames are matched against IP6_STD entries. If S2_IP6_CFG is set to 2, IPv6 TCP and UDP frames are matched against IP4_TCP_UDP entries. If S2_IP6_CFG is set to 3, IPv6 TCP and UDP frames are matched against MAC_ETYPE entries. **Note:** S2_IP6_CFG also controls the keys generation for IPv6 Other frames.

TABLE 2-34: HIERARCHY OF IS2 ENTRY TYPES (CONTINUED)

Frame Type	Description
IPv6 Other frames (non-TCP and non-UDP)	Configuration: S2_IP6_CFG. If S2_IP6_CFG is set to 0, IPv6 Other frames are matched against IP6_OTHER entries. If S2_IP6_CFG is set to 1, IPv6 Other frames are matched against IP6_STD entries. If S2_IP6_CFG is set to 2, IPv6 Other frames are matched against IP4_OTHER entries. If S2_IP6_CFG is set to 3, IPv6 Other frames are matched against MAC_ETYPE entries. **Note:** S2_IP6_CFG also controls the keys generation for IPv6 TCP and UDP frames.
IPv4 TCP and UDP frames	Configuration: S2_IP_TCPUDP_DIS If S2_IP_TCPUDP_DIS is cleared, IPv4 TCP and UDP frames are matched against IP4_TCPUDP entries. If S2_IP_TCPUDP_DIS is set, IPv4 TCP and UDP frames are matched against MAC_ETYPE entries.
IPv4 Other frames (non-TCP and non-UDP)	Configuration: S2_IP_OTHER_DIS If S2_IP_OTHER_DIS is cleared, IPv4 Other frames are matched against IP4_OTHER entries. If S2_IP_OTHER_DIS is set, IPv4 Other frames are matched against MAC_ETYPE entries.
ARP frames	Configuration: S2_ARP_DIS If S2_ARP_DIS is cleared, ARP frames are matched against ARP entries. If S2_ARP_DIS is set, ARP frames are matched against MAC_ETYPE entries.
OAM frames	Configuration: S2_OAM_DIS If S2_OAM_DIS is cleared, OAM frames are matched against OAM entries. If S2_OAM_DIS is set, OAM frames are matched against MAC_ETYPE entries.
SNAP frames	Configuration: S2_SNAP_DIS If S2_SNAP_DIS is cleared, SNAP frames are matched against SNAP entries. If S2_SNAP_DIS is set, SNAP frames are matched against LCC entries.

2.8.1.3 VCAP ES0 Port Configuration

The rewriter configures VCAP ES0 through REW:PORT:PORT_CFG.ES0_ENA. If ES0 is disabled, frames transmitted on the port are not matched against rules in ES0.

2.8.2 VCAP IS1

This section provides information about the IS1 keys and associated actions.

IS1 supports six different keys. The main characteristics of these keys are listed in the following table.

TABLE 2-35: OVERVIEW OF IS1 KEYS

IS1 Key	Size	Frame Types	Key Overview
S1_NORMAL	Half	Applicable to all frame types	Source MAC address, source IP address (32 bits) outer VLAN, DSCP, IP protocol, source and destination TCP/UDP ports.

TABLE 2-35: OVERVIEW OF IS1 KEYS (CONTINUED)

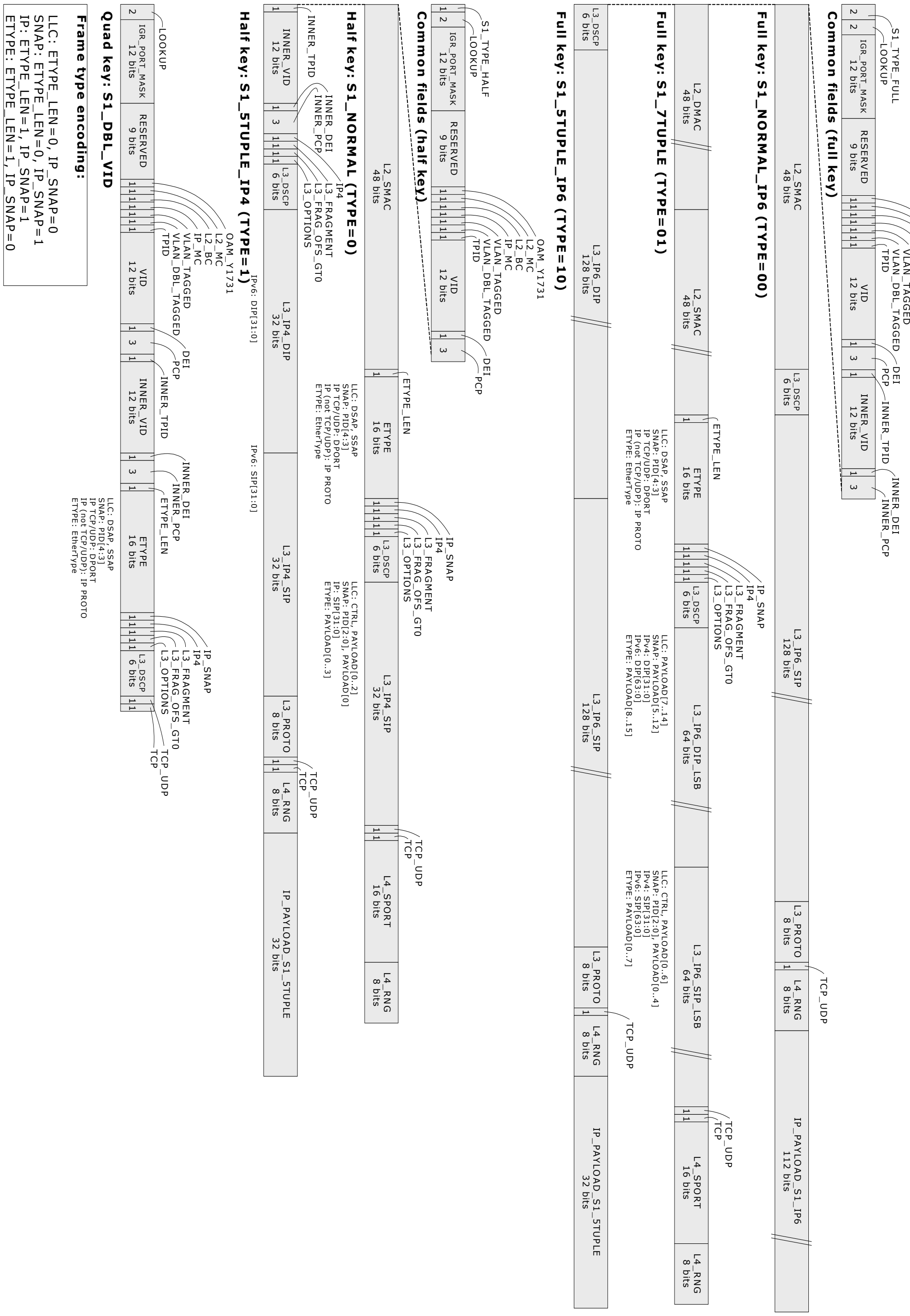
IS1 Key	Size	Frame Types	Key Overview
S1_5TUPLE_IP4	Half	Applicable to IPv4 and IPv6 frames	Inner and outer VLAN, source and destination IP addresses (32 bits), DSCP, IP protocol, source and destination TCP/UDP ports.
S1_NORMAL_IP6	Full	Applicable to IPv6 frames only	Similar to S1_NORMAL but with full IPv6 source IP address: Source MAC address, source IPv6 address (128 bits), inner and outer VLAN, DSCP, IP protocol, source and destination TCP/UDP ports.
S1_7TUPLE	Full	Applicable to all frame types	Source and destination MAC addresses, inner and outer VLAN, source and destination IP addresses (64 bits), DSCP, IP protocol, source and destination TCP/UDP ports.
S1_5TUPLE_IP6	Full	Applicable to IPv6 frames only	Similar to S1_5TUPLE_IP4 but with full IPv6 addresses: Inner and outer VLAN, source and destination IP addresses (128 bits), DSCP, IP protocol, source and destination TCP/UDP ports.
S1_DBL_VID	Quad	Applicable to all frame types	Subset of S1_7TUPLE without addresses: Inner and outer VLAN, DSCP, IP protocol, Destination TCP/UDP port.

2.8.2.1 IS1 Entry Key Encoding

All frame types are subject to the three IS1 lookups. The VCAP IS1 port configuration (VCAP_S1_CFG) determines the key generated for each lookup. Keys that are applicable to multiple frame types (for instance S1_NORMAL) contain frame type flags inside the key that indicate the originating frame type. In addition, certain key fields are overloaded with different frame fields depending on the frame type flag settings.

The following illustration shows the entry fields are available for quad, half, and full keys in IS1.

FIGURE 2-18: IS1 ENTRY TYPE OVERVIEW



VCAP IS1 can hold the following number of IS1 entries:

- S1_NORMAL: Up to 128 entries.
- S1_5TUPLE_IP4: Up to 128 entries.
- S1_NORMAL_IP6: Up to 64 entries.
- S1_7TUPLE: Up to 64 entries.
- S1_5TUPLE_IP6: Up to 64 entries.
- S1_DBL_VID: Up to 256 entries.

The following table provides information about how the quad key, S1_DBL_VID is generated.

TABLE 2-36: SPECIFIC FIELDS FOR IS1 QUAD KEY S1_DBL_VID

Field Name	Bit	Width	Description
Lookup Information			
LOOKUP	0	2	0: First lookup 1: Second lookup 2: Third lookup
Interface and Miscellaneous Information			
IGR_PORT_MASK	2	12	Ingress port mask. VCAP generated with one bit set in the mask corresponding to the ingress port.
RESERVED	14	9	Reserved. Must be set to don't care.
OAM_Y1731	23	1	Set if frame is a Y.1731 OAM frame with EtherType = 0x8902.
L2_MC	24	1	Set if frame's destination MAC address is a multicast address (bit 40 = 1)
L2_BC	15	1	Set if frame's destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF)
IP_MC	26	1	Set if frame is IPv4 frame and frame's destination MAC address is an IPv4 multicast address (0x01005E0 /25). Set if frame is IPv6 frame and frame's destination MAC address is an IPv6 multicast address (0x3333/16).
Tagging Information			
VLAN_TAGGED	27	1	Set if frame has one or more Q-tags. Independent of port VLAN awareness.
VLAN_DBL_TAGGED	28	1	Set if frame has two or more Q-tags. Independent of port VLAN awareness.
TPID	29	1	0: Customer TPID 1: Service TPID (88A8 or programmable). S1_NORMAL: TPID is derived from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA.
VID	30	12	Frame's VID if frame is tagged, otherwise port default. S1_NORMAL: VID is taken from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA.
DEI	42	1	Frame's DEI if frame is tagged, otherwise port default. S1_NORMAL: DEI is taken from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA.
PCP	43	3	Frame's PCP if frame is tagged, otherwise port default. S1_NORMAL: PCP is taken from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA.
INNER_TPID	46	1	TPID for inner tag: 0: Customer TPID. 1: Service TPID (88A8 or programmable).
INNER_VID	47	12	Frame's inner VID if frame is double tagged.
INNER_DEI	59	1	Frame's inner DEI if frame is double tagged.
INNER_PCP	60	3	Frame's inner PCP if frame is double tagged.
Layer 2 Information			
ETYPE_LEN	63	1	Frame type flag. Set if frame has EtherType >= 0x600 (Frame is type encoded). Otherwise cleared (Frame is length encoded).

TABLE 2-36: SPECIFIC FIELDS FOR IS1 QUAD KEY S1_DBL_VID (CONTINUED)

Field Name	Bit	Width	Description
ETYPE	64	16	Overloaded field for different frame types: LLC frame: ETYPE = [DSAP, SSAP] SNAP frame: ETYPE = PID[4:3] IPv4 or IPv6 TCP/UDP frame: ETYPE = DPORT IPv4 or IPv6 Other frame: ETYPE = IP protocol ARP or ETYPE frame: ETYPE = Frame's EtherType. Y.1731 OAM frames (EtherType = 0x8902): ETYPE[15:8] = 0x89. ETYPE[7]: Set if the frames was injected with masquerading. ETYPE[6:0]: Encodes the frame's MEG level. For more information, see OAM_MEL_FLAGS Table 2-53. To indicate the overloading of the ETYPE field, OAM_Y1731 is set to 1.
IP_SNAP	80	1	Frame type flag. Set if frame is IPv4, IPv6, or SNAP frame.
IP4	81	1	Frame type flag. Set if frame is IPv4 frame
Layer 3 Information			
L3_FRAGMENT	82	1	Set if IPv4 frame is fragmented (More Fragments flag = 1 or Fragments Offset > 0). Layer 4 information cannot not be trusted.
L3_-FRAG_OFS_GT0	83	1	Set if IPv4 frame is fragmented and it is not the first fragment (Fragments Offset > 0). Layer 4 information cannot not be trusted.
L3_OPTIONS	84	1	Set if IPv4 frame contains options (IP len > 5). IP options are not skipped nor parsed. Layer 4 information cannot not be trusted.
L3_DSCP	85	6	Frame's DSCP value. The DSCP value may have been translated during basic classification. See Section 2.7.3, QoS, DP, and DSCP Classification.
Layer 4 Information			
TCP_UDP	91	1	Frame type flag. Set if frame is IPv4/IPv6 TCP or UDP frame.
TCP	92	1	Frame type flag. Set if frame is IPv4/IPv6 TCP frame.

The following three tables provide information about how the half keys, S1_NORMAL and S1_5TUPLE_IP4, are generated. The first table lists the common fields between the half keys.

TABLE 2-37: IS1 COMMON KEY FIELDS FOR HALF KEYS

Field Name	Bit	Width	Description
Lookup Information			
IS1_TYPE_HALF	0	1	0: S1_NORMAL entries. 1: S1_5TUPLE_IP4 entries.
LOOKUP	1	2	0: First lookup. 1: Second lookup. 2: Third lookup.
Interface and Miscellaneous Information			
IGR_PORT_MASK	3	12	Ingress port mask. VCAP generated with one bit set in the mask corresponding to the ingress port.
RESERVED	15	9	Reserved. Must be set to don't care.

TABLE 2-37: IS1 COMMON KEY FIELDS FOR HALF KEYS (CONTINUED)

Field Name	Bit	Width	Description
OAM_Y1731	24	1	Set if frame is Y.1731 OAM frame with EtherType = 0x8902.
L2_MC	25	1	Set if frame's destination MAC address is a multicast address (bit 40 = 1).
L2_BC	26	1	Set if frame's destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF)
IP_MC	27	1	Set if frame is IPv4 frame and frame's destination MAC address is an IPv4 multicast address (0x01005E0 /25). Set if frame is IPv6 frame and frame's destination MAC address is an IPv6 multicast address (0x3333/16).
Tagging Information			
VLAN_TAGGED	28	1	Set if frame has one or more Q-tags. Independent of port VLAN awareness.
VLAN_DBL_TAGGED	29	1	Set if frame has two or more Q-tags. Independent of port VLAN awareness.
TPID	30	1	0: Customer TPID. 1: Service TPID (88A8 or programmable). S1_NORMAL: TPID is derived from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA.
VID	31	12	Frame's VID if frame is tagged, otherwise port default. S1_NORMAL: VID is taken from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA.
DEI	43	1	Frame's DEI if frame is tagged, otherwise port default. S1_NORMAL: DEI is taken from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA.
PCP	44	3	Frame's PCP if frame is tagged, otherwise port default. S1_NORMAL: PCP is taken from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA.

TABLE 2-38: SPECIFIC FIELDS FOR IS1 HALF KEY S1_NORMAL

Field Name	Bit	Width	Description
Layer-2 Information			
L2_SMAC	47	48	Frame's source MAC address. Use destination MAC address if VCAP_CFG.S1_DMAC_DIP_ENA is set for ingress port.
ETYPE_LEN	95	1	Frame type flag. Set if frame has EtherType >= 0x600 (Frame is type encoded). Otherwise cleared (Frame is length encoded).
ETYPE	96	16	Overloaded field for different frame types: LLC frame: ETYPE = [DSAP, SSAP] SNAP frame: ETYPE = PID[4:3] IPv4 or IPv6 TCP/UDP frame: ETYPE = DPORT IPv4 or IPv6 Other frame: ETYPE = IP protocol ARP or ETYPE frame: ETYPE = Frame's EtherType. Y.1731 OAM frames (EtherType = 0x8902): ETYPE[15:8] = 0x89. ETYPE[7]: Set if the frames was injected with masquerading. ETYPE[6:0]: Encodes the frame's MEG level. For more information, see OAM_MEL_FLAGS Table 2-53. To indicate the overloading of the ETYPE field, OAM_Y1731 is set to 1.
IP_SNAP	112	1	Frame type flag. Set if frame is IPv4, IPv6, or SNAP frame.

TABLE 2-38: SPECIFIC FIELDS FOR IS1 HALF KEY S1_NORMAL (CONTINUED)

Field Name	Bit	Width	Description
IP4	113	1	Frame type flag. Set if frame is IPv4 frame
Layer-3 Information			
L3_FRAGMENT	114	1	Set if IPv4 frame is fragmented (More Fragments flag = 1 or Fragments Offset > 0). Layer 4 information cannot not be trusted.
L3_-FRAG_OFS_GT0	115	1	Set if IPv4 frame is fragmented and it is not the first fragment (Fragments Offset > 0). Layer 4 information cannot not be trusted.
L3_OPTIONS	116	1	Set if IPv4 frame contains options (IP len > 5). IP options are not skipped nor parsed. Layer 4 information cannot not be trusted.
L3_DSCP	117	6	Frame's DSCP value. The DSCP value may have been translated during basic classification. See Section 2.7.3, QoS, DP, and DSCP Classification.
L3_IP4_SIP	123	32	Overloaded fields for different frame types: LLC frame: L3_IP4_SIP = [CTRL, PAYLOAD[0:2]] SNAP frame: L3_IP4_SIP = [PID[2:0], PAYLOAD[0]] IPv4 or IPv6 frame: L3_IP4_SIP = source IP address, bits [31:0] OAM, ARP or ETYPE frame: L3_IP4_SIP = PAYLOAD[0:3] For IPv4 or IPv6 frames, use destination IP address if VCAP_CFG.S1_DMAC_DIP_ENA is set for ingress port.
Layer-4 Information			
TCP_UDP	155	1	Frame type flag. Set if frame is IPv4/IPv6 TCP or UDP frame.
TCP	156	1	Frame type flag. Set if frame is IPv4/IPv6 TCP frame.
L4_SPORT	157	16	TCP/UDP frame's source port.
L4_RNG	173	8	Range mask with one bit per range. A bit is set, if the corresponding range is matched. Range types: SPORT, DPORT, SPORT or DPORT, VID, DSCP Input to range checkers: SPORT/DPORT: From frame VID: From frame if tagged, otherwise port's VID DSCP: Translated DSCP from the basic classification. See Section 2.8.5, Range Checkers.

TABLE 2-39: SPECIFIC FIELDS FOR IS1 HALF KEY S1_5TUPLE_IP4

Field Name	Bit	Width	Description
Tagging Information			
INNER_TPID	47	1	TPID for inner tag. 0: Customer TPID. 1: Service TPID (88A8 or programmable).
INNER_VID	48	12	Frame's inner VID if frame is double tagged.
INNER_DEI	60	1	Frame's inner DEI if frame is double tagged.
INNER_PCP	61	3	Frame's inner PCP if frame is double tagged.
Layer-3 Information			
IP4	64	1	Frame type flag. Set if frame is IPv4 frame.

TABLE 2-39: SPECIFIC FIELDS FOR IS1 HALF KEY S1_5TUPLE_IP4 (CONTINUED)

Field Name	Bit	Width	Description
L3_FRAGMENT	65	1	Set if IPv4 frame is fragmented (More Fragments flag = 1 or Fragments Offset > 0). Layer 4 information cannot not be trusted.
L3_FRAG_OFS_GT0	66	1	Set if IPv4 frame is fragmented and it is not the first fragment (Fragments Offset > 0). Layer 4 information cannot not be trusted.
L3_OPTIONS	67	1	Set if IPv4 frame contains options (IP len > 5). IP options are not skipped nor parsed. Layer 4 information cannot not be trusted.
L3_DSCP	68	6	Frame's DSCP value. The DSCP value may have been translated during basic classification. For more information, see Section 2.7.3, QoS, DP, and DSCP Classification.
L3_IP4_DIP	74	32	IPv4: destination IP address. IPv6: destination IP address, bits [31:0].
L3_IP4_SIP	106	32	IPv4: source IP address. IPv6: source IP address, bits [31:0].
L3_PROTO	138	8	IPv4: IP protocol. IPv6: next header.
Layer-4 Information			
TCP_UDP	146	1	Frame type flag. Set if frame is IPv4/IPv6 TCP or UDP frame.
TCP	147	1	Frame type flag. Set if frame is IPv4/IPv6 TCP frame.
L4_RNG	148	8	Range mask with one bit per range. A bit is set, if the corresponding range is matched. Range types: SPORT, DPORT, SPORT or DPORT, VID, DSCP Input to range checkers: SPORT/DPORT: From frame VID: From frame if tagged, otherwise port's VID DSCP: Translated DSCP from the basic classification. See Section 2.8.5, Range Checkers.
IP_PAYLOAD_S1_5TUPLE	156	32	Payload after IPv4 or IPv6 header. For TCP and UDP frames, this field contains the source and destination port numbers.

The following four tables provide information about how the full keys, S1_NORMAL_IP6, S1_7TUPLE, and S1_5TUPLE_IP6, are generated. The first table lists the common fields between the full keys.

TABLE 2-40: IS1 COMMON KEY FIELDS FOR FULL KEYS

Field Name	Bit	Width	Description
Lookup Information			
IS1_TYPE_FULL	0	2	0: S1_NORMAL_IP6 entries. 1: S1_7TUPLE entries. 2: S1_5TUPLE_IP6 entries.
LOOKUP	2	2	0: First lookup. 1: Second lookup. 2: Third lookup.
Interface and Miscellaneous Information			

TABLE 2-40: IS1 COMMON KEY FIELDS FOR FULL KEYS (CONTINUED)

Field Name	Bit	Width	Description
IGR_PORT_MASK	4	12	Ingress port mask. VCAP generated with one bit set in the mask corresponding to the ingress port.
RESERVED	16	9	Reserved. Must be set to don’t care.
OAM_Y1731	25	1	Set if frame is a Y.1731 frame with EtherType = 0x8902.
L2_MC	26	1	Set if frame’s destination MAC address is a multicast address (bit 40 = 1).
L2_BC	27	1	Set if frame’s destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF).
IP_MC	28	1	Set if frame is IPv4 frame and frame’s destination MAC address is an IPv4 multicast address (0x01005E0 /25). Set if frame is IPv6 frame and frame’s destination MAC address is an IPv6 multicast address (0x3333/16).
Tagging Information			
VLAN_TAGGED	29	1	Set if frame has one or more Q-tags. Independent of port VLAN awareness.
VLAN_DBL_TAGGED	30	1	Set if frame has two or more Q-tags. Independent of port VLAN awareness.
TPID	31	1	0: Customer TPID. 1: Service TPID (88A8 or programmable). S1_NORMAL: TPID is derived from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA.
VID	32	12	Frame’s VID if frame is tagged, otherwise port default. S1_NORMAL: VID is taken from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA.
DEI	44	1	Frame’s DEI if frame is tagged, otherwise port default. S1_NORMAL: DEI is taken from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA.
PCP	45	3	Frame’s PCP if frame is tagged, otherwise port default. S1_NORMAL: PCP is taken from tag pointed to by VCAP_CFG.S1_VLAN_INNER_TAG_ENA.
INNER_TPID	48	1	TPID for inner tag. 0: Customer TPID. 1: Service TPID (88A8 or programmable).
INNER_VID	49	12	Frame’s inner VID if frame is double tagged.
INNER_DEI	61	1	Frame’s inner DEI if frame is double tagged.
INNER_PCP	62	3	Frame’s inner PCP if frame is double tagged.

TABLE 2-41: SPECIFIC FIELDS FOR IS1 FULL KEY S1_NORMAL_IP6

Field Name	Bit	Width	Description
Layer 2 Information			
L2_SMAC	65	48	Frame’s source MAC address. Use destination MAC address if VCAP_CFG.S1_DMAC_DIP_ENA is set for ingress port.
Layer 3 Information			
L3_DSCP	113	6	Frame’s DSCP value. The DSCP value may have been translated during basic classification. For more information, see Section 2.7.3, QoS, DP, and DSCP Classification.

TABLE 2-41: SPECIFIC FIELDS FOR IS1 FULL KEY S1_NORMAL_IP6 (CONTINUED)

Field Name	Bit	Width	Description
L3_IP6_SIP	119	128	Source IP address. Use destination IP address if VCAP_CFG.S1_DMAC_DIP_ENA is set for ingress port.
L3_PROTO	247	8	IPv6 next header.
Layer 4 Information			
TCP_UDP	255	1	Frame type flag. Set if frame is IPv4/IPv6 TCP or UDP frame.
TCP	256	1	Frame type flag. Set if frame is IPv4/IPv6 TCP frame.
L4_RNG	257	8	Range mask with one bit per range. A bit is set, if the corresponding range is matched. Range types: SPORT, DPORT, SPORT or DPORT, VID, DSCP. Input to range checkers: SPORT/DPORT: From frame VID: From frame if tagged, otherwise port's VID DSCP: Translated DSCP from the basic classification. See Section 2.8.5, Range Checkers.
IP_PAYLOAD_S1_IP6	265	112	Payload after IPv6 header. For TCP and UDP frames, this field contains the source and destination port numbers.

TABLE 2-42: SPECIFIC FIELDS FOR IS1 FULL KEY S1_7TUPLE

Field Name	Bit	Width	Description
Layer-2 Information			
L2_DMAC	65	48	Frame's destination MAC address.
L2_SMAC	113	48	Frame's source MAC address.
ETYPE_LEN	161	1	Frame type flag. Set if frame has EtherType >= 0x600 (Frame is type encoded). Otherwise cleared (Frame is length encoded).
ETYPE	162	16	Overloaded field for different frame types: LLC frame: ETYPE = [DSAP, SSAP] SNAP frame: ETYPE = PID[4:3] IPv4 or IPv6 TCP/UDP frame: ETYPE = DPORT IPv4 or IPv6 Other frame: ETYPE = IP protocol ARP or ETYPE frame: ETYPE = Frame's EtherType. Y.1731 OAM frames (EtherType = 0x8902): ETYPE[15:8] = 0x89. ETYPE[7]: Set if the frames was injected with masquerading. ETYPE[6:0]: Encodes the frame's MEG level. For more information, see OAM_MEL_FLAGS Table 2-53. To indicate the overloading of the ETYPE field, OAM_Y1731 is set to 1.
IP_SNAP	178	1	Frame type flag. Set if frame is IPv4, IPv6, or SNAP frame.
IP4	179	1	Frame type flag. Set if frame is IPv4 frame
Layer-3 Information			
L3_FRAGMENT	180	1	Set if IPv4 frame is fragmented (More Fragments flag = 1 or Fragments Offset > 0). Layer 4 information cannot not be trusted.

TABLE 2-42: SPECIFIC FIELDS FOR IS1 FULL KEY S1_7TUPLE (CONTINUED)

Field Name	Bit	Width	Description
L3_FRAG_OFS_GT0	181	1	Set if IPv4 frame is fragmented and it is not the first fragment (Fragments Offset > 0). Layer 4 information cannot not be trusted.
L3_OPTIONS	182	1	Set if IPv4 frame contains options (IP len > 5). IP options are not skipped nor parsed. Layer 4 information cannot not be trusted.
L3_DSCP	183	6	Frame's DSCP value. The DSCP value may have been translated during basic classification. For more information, see Section 2.7.3, QoS, DP, and DSCP Classification.
L3_IP6_DIP_MSB	189	16	IPv6 frame: L3_IP6_SIP_MSB = destination IP address, bits [127:112]
L3_IP6_DIP	205	64	Overloaded fields for different frame types: LLC frame: L3_IP6_DIP = PAYLOAD[7:14] SNAP frame: L3_IP6_DIP = PAYLOAD[5:12] IPv4 frame: L3_IP6_DIP = destination IP address, bits [31:0] IPv6 frame: L3_IP6_DIP = destination IP address, bits [63:0] OAM, ARP or ETYPE frame: L3_IP6_DIP = PAYLOAD[8:15]
L3_IP6_SIP_MSB	269	16	IPv6 frame: L3_IP6_SIP_MSB = source IP address, bits [127:112]
L3_IP6_SIP	285	64	Overloaded fields for different frame types: LLC frame: L3_IP6_SIP = [CTRL, PAYLOAD[0:6]] SNAP frame: L3_IP6_SIP = [PID[2:0], PAYLOAD[0:4]] IPv4 frame: L3_IP6_SIP = source IP address, bits [31:0] IPv6 frame: L3_IP6_SIP = source IP address, bits [63:0] OAM, ARP or ETYPE frame: L3_IP6_SIP = PAYLOAD[0:7]
Layer-4 Information			
TCP_UDP	349	1	Frame type flag. Set if frame is IPv4/IPv6 TCP or UDP frame.
TCP	350	1	Frame type flag. Set if frame is IPv4/IPv6 TCP frame.
L4_SPORT	351	16	TCP/UDP frame's source port.
L4_RNG	367	8	Range mask with one bit per range. A bit is set, if the corresponding range is matched. Range types: SPORT, DPORT, SPORT or DPORT, VID, DSCP Input to range checkers: SPORT/DPORT: From frame VID: From frame if tagged, otherwise port's VID DSCP: Translated DSCP from the basic classification. See Section 2.8.5, Range Checkers.

TABLE 2-43: SPECIFIC FIELDS FOR IS1 FULL KEY S1_5TUPLE_IP6

Field Name	Bit	Width	Description
Layer-3 Information			
L3_DSCP	65	6	Frame's DSCP value. The DSCP value may have been translated during basic classification. See Section 2.7.3, QoS, DP, and DSCP Classification.
L3_IP6_DIP	71	128	IPv6 destination IP address
L3_IP6_SIP	199	128	IPv6 source IP address
L3_PROTO	327	8	IPv6 next header.
Layer-4 Information			

TABLE 2-43: SPECIFIC FIELDS FOR IS1 FULL KEY S1_5TUPLE_IP6 (CONTINUED)

Field Name	Bit	Width	Description
TCP_UDP	335	1	Frame type flag. Set if frame is IPv6 TCP or UDP frame.
TCP	336	1	Frame type flag. Set if frame is IPv6 TCP frame.
L4_RNG	337	8	Range mask with one bit per range. A bit is set, if the corresponding range is matched. Range types: SPORT, DPORT, SPORT or DPORT, VID, DSCP Input to range checkers: SPORT/DPORT: From frame VID: From frame if tagged, otherwise port's VID DSCP: Translated DSCP from the basic classification. See Section 2.8.5, Range Checkers.
IP_PAYLOAD_S1_5TUPLE	345	32	Payload after IPv6 header. For TCP and UDP frames, this field contains the source and destination port numbers.

Fields not applicable to a certain frame type (for example, L3_OPTIONS for an IPv6 frame) must be set to don't care for entries the frame type can match.

If L3_FRAGMENT or L3_OPTIONS are set to 1 or set to don't care, Layer 4 information cannot be trusted and should be set to don't-care for such entries.

2.8.2.2 IS1 Action Encoding

The VCAP generates an action vector from each of the three IS1 lookups. All matches return the same action vector, independently of the match entry type. The action vectors are combined into one action vector, which is applied to the classification of the frame.

There are no default action vectors for the IS1.

The following table lists the available fields for the IS1 action vector.

TABLE 2-44: IS1 ACTION FIELDS

Action Field	Bit	Width	Description
DSCP_ENA	0	1	If set, use DSCP_VAL as classified DSCP value. Otherwise, DSCP value from basic classification is used.
DSCP_VAL	1	6	See DSCP_ENA.
QOS_ENA	7	1	If set, use QOS_VAL as classified QoS class. Otherwise, QoS class from basic classification is used.
QOS_VAL	8	3	See QOS_ENA.
DP_ENA	11	1	If set, use DP_VAL as classified drop precedence level. Otherwise, drop precedence level from basic classification is used.
DP_VAL	12	1	See DP_ENA.
PAG_OVERRIDE_MASK	13	8	Bits set in this mask will override PAG_VAL from port profile. New PAG = (PAG (input) AND ~PAG_OVERRIDE_MASK) OR (PAG_VAL AND PAG_OVERRIDE_MASK)
PAG_VAL	21	8	See PAG_OVERRIDE_MASK.
RESERVED	29	9	Reserved. Must be set to 0.

TABLE 2-44: IS1 ACTION FIELDS (CONTINUED)

Action Field	Bit	Width	Description
VID_REPLACE_ENA	40	1	Controls the classified VID: VID_REPLACE_ENA=0: Add VID_ADD_VAL to basic classified VID and use result as new classified VID. VID_REPLACE_ENA = 1: Replace basic classified VID with VID_VAL value and use as new classified VID.
VID_ADD_VAL	41	12	See VID_REPLACE_ENA.
FID_SEL	53	2	Controls the Filter Identifier (FID) used when looking up the MAC table. 0: Disabled: FID = classified VID prepended 0. 1: Use FID_VAL for SMAC lookup in MAC table. 2: Use FID_VAL for DMAC lookup in MAC table. 3: Use FID_VAL for DMAC and SMAC lookup in MAC table.
FID_VAL	55	13	See FID_SEL.
PCP_DEI_ENA	68	1	If set, use PCP_VAL and DEI_VAL as classified PCP and DEI values. Otherwise, PCP and DEI from basic classification are used.
PCP_VAL	69	3	See PCP_DEI_ENA.
DEI_VAL	72	1	See PCP_DEI_ENA.
VLAN_POP_CNT_ENA	73	1	If set, use VLAN_POP_CNT as the number of VLAN tags to pop from the incoming frame. This number is used by the Rewriter. Otherwise, VLAN_POP_CNT from ANA:PORT:VLAN_CFG.VLAN_POP_CNT is used.
VLAN_POP_CNT	74	2	See VLAN_POP_CNT_ENA.
CUSTOM_ACE_TYPE_ENA	76	4	Enables use of custom keys in IS2. Bits 3:2 control second lookup in IS2 while bits 1:0 control first lookup. Encoding per lookup: 0: Disabled. 1: Extract 40 bytes after position corresponding to the location of the IPv4 header and use as key. 2: Extract 40 bytes after SMAC and use as key.
HIT_STICKY	80	1	If set, a frame has matched against the associated entry.

Each lookup returns an action vector if there is a match. The potentially three IS1 action vectors are applied in three steps. First, the action vector from the first lookup is applied, then the action vector from the second lookup is applied to the result from the first action vector, and finally, the action vector from the third lookup is applied to the result from the second action vector. This implies that if two or more lookups return an action of DP_ENA = 1; for example, the DP_VAL from the last lookup is used.

The CUSTOM_ACE_TYPE_ENA action in IS1 enables the use of custom keys in IS2. Two different custom keys are supported:

- CUSTOM_IP: 40 bytes are extracted from the frame starting from byte-position 34. This position corresponds to the first byte after the IPv4 header if the frame is IPv4. The 40 bytes are matched against the CUSTOM entries in IS2.
- CUSTOM_SMAC: 40 bytes are extracted from the frame starting from byte-position 12. This position corresponds to the first byte after the source MAC address. The 40 bytes are matched against the CUSTOM entries in IS2.

Note Up to two VLAN tags (two for double VLAN tagged frames and one for single VLAN tagged frames) are removed from the frame before extracting the data.

2.8.3 VCAP IS2

This section provides information about the IS2 keys, the SMAC_SIP4 and SMAC_SIP6 keys, and associated actions.

2.8.3.1 IS2 Entry Key Encoding

All frame types are subject to the two IS2 lookups. The frame type determines the key entry type. For more information about VCAP frame types, see Table 2-32. The following illustrations show entry fields available for each frame type (indicated by the field IS2_TYPE_HALF and IS2_TYPE_FULL) for half and full entries.

FIGURE 2-19: IS2 HALF ENTRY TYPE OVERVIEW

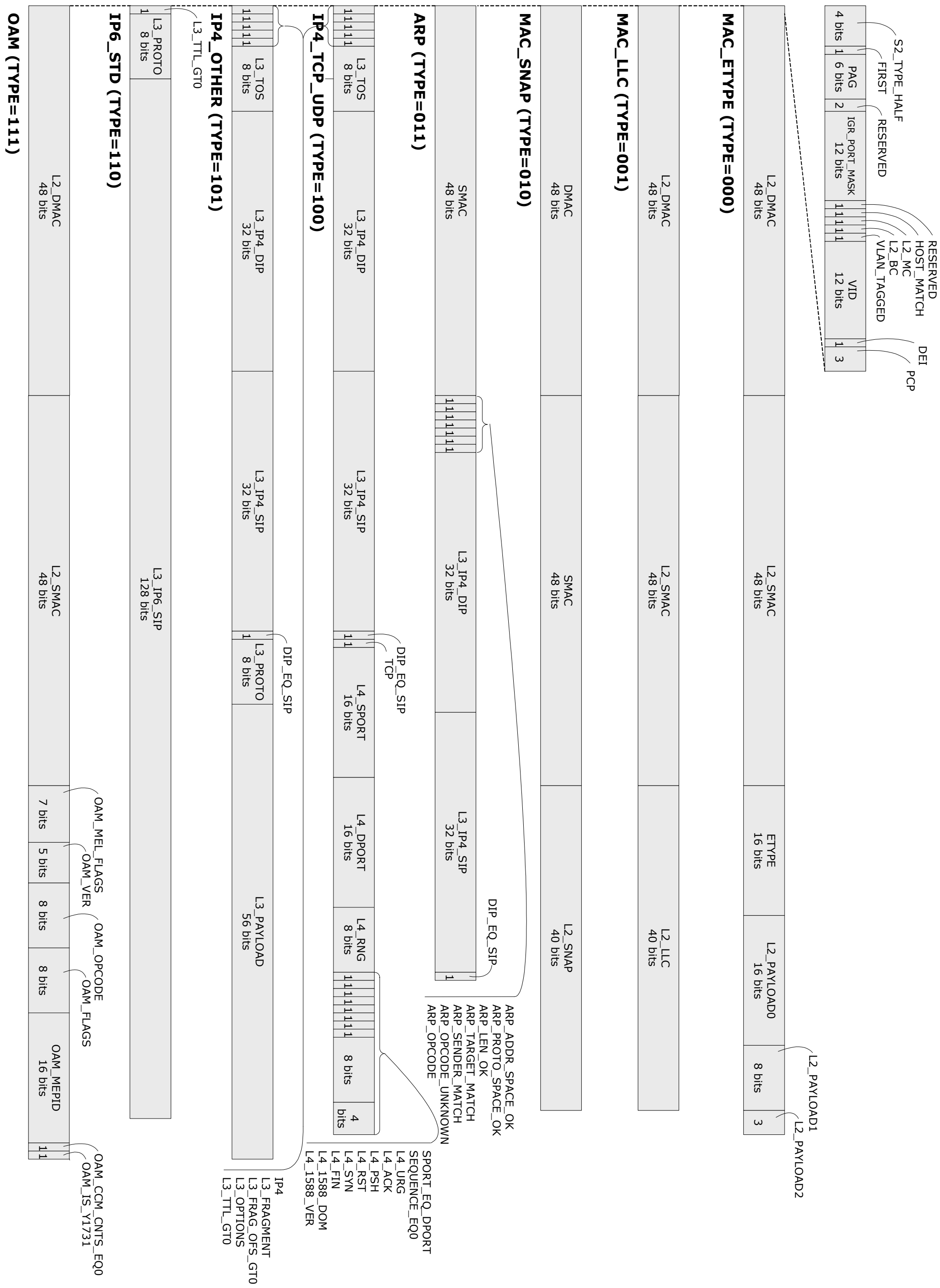
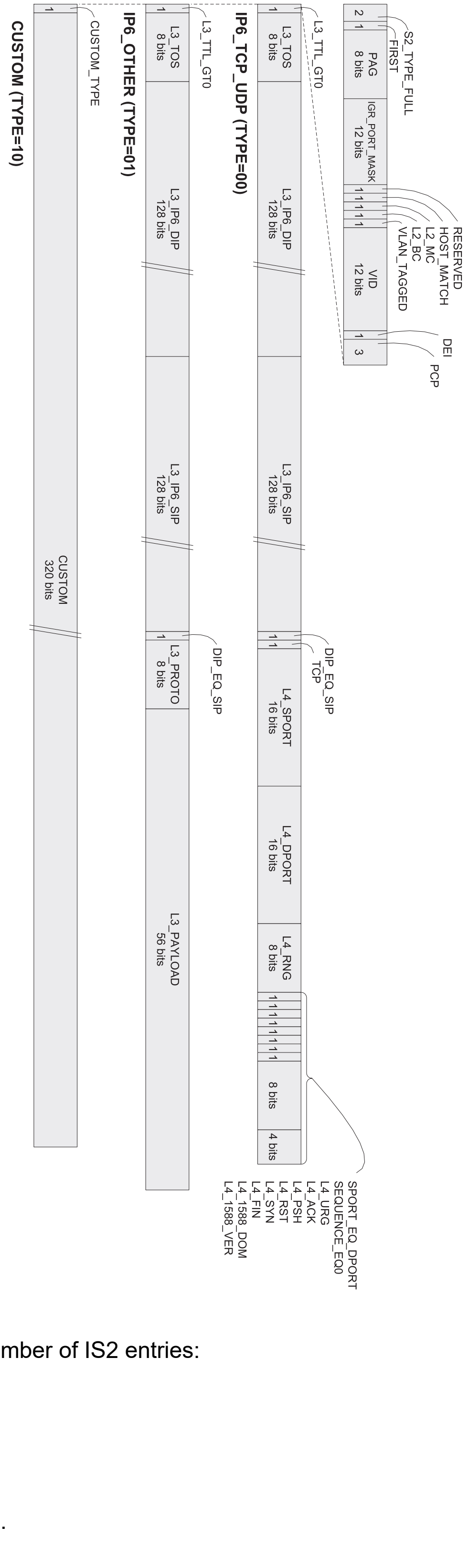


FIGURE 2-20: IS2 HALF ENTRY TYPE OVERVIEW



VCAP IS2 can hold the following number of IS2 entries:

- MAC_ETYPE: Up to 128 entries.
- MAC_LLC: Up to 128 entries.
- MAC_SNAP: Up to 128 entries.
- ARP: Up to 128 entries.
- IP4_TCP_UDP: Up to 128 entries.
- IP4_OTHER: Up to 128 entries.
- IP6_STD: Up to 128 entries.

- IP6_TCP_UDP: Up to 64 entries.
- IP6_OTHER: Up to 64 entries.
- OAM: Up to 128 entries.

The following tables provide information about how the IS2 half keys are generated. The first table lists the common fields between the half keys.

TABLE 2-45: IS2 COMMON KEY FIELDS FOR HALF KEYS

Field Name	Bit	Width	Description
Lookup Information			
S2_TYPE_HALF	0	4	0: MAC ETYPE entries 1: MAC LLC entries 2: MAC SNAP entries 3: ARP entries 4: IPv4 TCP/UDP entries 5: IPv4 OTHER entries 6: IPv6 STD entries 7: OAM entries (8: SMAC_SIP6 entries)
FIRST	4	1	Set for first lookup and cleared for second lookup.
Interface Information			
PAG	5	8	Policy association group. Action from VCAP IS1.
IGR_PORT_MASK	13	12	Ingress port mask. VCAP generated with one bit set in the mask corresponding to the ingress port.
Tagging and IP Source Guarding Information			
RESERVED	26	1	Reserved. Must be set to don’t care.
HOST_MATCH	26	1	The action from the SMAC_SIP4 or SMAC_SIP6 lookups. Used for IP source guarding.
L2_MC	27	1	Set if frame’s destination MAC address is a multicast address (bit 40 = 1).
L2_BC	28	1	Set if frame’s destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF).
VLAN_TAGGED	29	1	Set if frame has one or more Q-tags. Independent of port VLAN awareness.
VID	30	12	Classified VID which is the result of the VLAN classification in basic classification and IS1.
DEI	42	1	Classified DEI which is the final result of the VLAN classification in basic classification and IS1.
PCP	43	3	Classified PCP which is the final result of the VLAN classification in basic classification and IS1.

TABLE 2-46: IS2 MAC_ETYPE KEY

Field Name	Bit	Width	Description
Layer-2 Information			
L2_DMAC	46	48	Frame’s destination MAC address.
L2_SMAC	94	48	Frame’s source MAC address.
ETYPE	142	16	Frame’s EtherType. This is the EtherType after up to two VLAN tags.
L2_PAYLOAD0	158	16	Payload bytes 0-1 after the frame’s EtherType.
L2_PAYLOAD1	174	8	Payload byte 4 after the frame’s EtherType. This is specifically for PTP frames.

TABLE 2-46: IS2 MAC_ETYPE KEY (CONTINUED)

Field Name	Bit	Width	Description
L2_PAYLOAD2	182	3	Bits 7, 2, and 1 from payload byte 6 after the frame's Ether-Type. This is specifically for PTP frames.

TABLE 2-47: IS2 MAC_LLC KEY

Field Name	Bit	Width	Description
Layer-2 Information			
L2_DMAC	46	48	Frame's destination MAC address
L2_SMAC	94	48	Frame's source MAC address
L2_LLC	142	40	LLC header and data after up to two VLAN tags and the type/length field.

TABLE 2-48: IS2 MAC_SNAP KEY

Field Name	Bit	Width	Description
Layer-2 Information			
L2_DMAC	46	48	Frame's destination MAC address
L2_SMAC	94	48	Frame's source MAC address
L2_SNAP	142	40	SNAP header after LLC header (AA-AA-03).

TABLE 2-49: IS2 ARP KEY

Field Name	Bit	Width	Description
Layer-2 Information			
L2_SMAC	46	48	Frame's source MAC address
Layer-3 Information			
ARP_ADDR_SPACE_OK	94	1	Set if hardware address is Ethernet.
ARP_PROTO_SPACE_OK	95	1	Set if protocol address space is IP.
ARP_LEN_OK	96	1	Set if hardware address length = 6 (Ethernet) and IP address length = 4 (IP).
ARP_TARGET_MATCH	97	1	Target hardware address = SMAC (RARP).
ARP_SENDER_MATCH	98	1	Sender hardware address = SMAC (ARP).
ARP_OPCODE_UNKNOWN	99	1	Set if ARP opcode is none of the below are mentioned.
ARP_OPCODE	100	2	0: ARP request 1: ARP reply. 2: RARP request. 3: RARP reply.
L3_IP4_DIP	102	32	Target IPv4 address.
L3_IP4_SIP	134	32	Sender IPv4 address.
DIP_EQ_SIP	166	1	Set if sender IP address is equal to target IP address.

TABLE 2-50: IS2 IP4_TCP_UDP KEY

Field Name	Bit	Width	Description
Layer-3 and Layer-4 Information			
IP4	46	1	Set if frame is IPv4 frame. IPv6 frames can also use IP4_TCP_UDP entries when IP6_STD entries are disabled.

TABLE 2-50: IS2 IP4_TCP_UDP KEY (CONTINUED)

Field Name	Bit	Width	Description
L3_FRAGMENT	47	1	Set if IP frame is fragmented (More Fragments flag = 1 or Fragments Offset > 0).
L3_FRAG_OFS_GT0	48	1	Set if IP frame is fragmented and it is not the first fragment (Fragments Offset > 0). Such frames do not carry Layer-4 information all Layer-4 information fields in the key are automatically set to don't-care when generating the key.
L3_OPTIONS	49	1	Set if IP frame contains options (IP len > 5). IP options are not skipped nor parsed which implies that Layer-4 information cannot be used. All Layer-4 information fields in the key are automatically set to don't-care when generating the key.
L3_TTL_GT0	50	1	Set if IP TTL is greater than 0.
L3_TOS	51	8	IP TOS field.
L3_IP4_DIP	59	32	IPv4 frames: Destination IPv4 address. IPv6 frames: Source IPv6 address, bits 63:32.
L3_IP4_SIP	91	32	IPv4 frames: Source IPv4 address. IPv6 frames: Source IPv6 address, bit 31:0.
DIP_EQ_SIP	123	1	Set if source IP address is equal to destination IP address. Full addresses are checked, also for IPv6.
TCP	124	1	Set if IP Proto = 6 (TCP).
L4_DPORT	125	16	TCP/UDP destination port.
L4_SPORT	141	16	TCP/UDP source port.
L4_RNG	157	8	Range mask with one bit per range. A bit is set, if the corresponding range is matched. Range types: SPORT, DPORT, SPORT or DPORT, VID, DSCP. Input to range checkers: SPORT, DPORT: From frame VID, DSCP: Classified result from IS1. See Section 2.8.5, Range Checkers.
SPORT_EQ_DPORT	165	1	Set if UDP or TCP source port equals UDP or TCP destination port.
SEQUENCE_EQ0	166	1	TCP: Set if TCP sequence number is 0. PTP over UDP: messageType bit 0.
L4_FIN	167	1	TCP: TCP flag FIN. PTP over UDP: messageType bit 1.
L4_SYN	168	1	TCP: TCP flag SYN. PTP over UDP: messageType bit 2.
L4_RST	169	1	TCP: TCP flag RST. PTP over UDP: messageType bit 3.
L4_PSH	170	1	TCP: TCP flag PSH. PTP over UDP: flagField bit 1 (twoStepFlag).
L4_ACK	171	1	TCP: TCP flag ACK. PTP over UDP: flagField bit 2 (unicastFlag).
L4_URG	172	1	TCP: TCP flag URG. PTP over UDP: flagField bit 7 (reserved).
L4_1588_DOM	173	8	PTP over UDP: domainNumber.
L4_1588_VERSION	181	4	PTP over UDP: version.

Frames with IP options (L3_OPTIONS set to 1 in key) or fragmented frames, which are not the initial fragment (L3_-FRAG_OFS_GT0 set to 1 in key), do not carry Layer-4 information. The Layer-4 fields in the key (L4_SPORT, L4_D-PORT, L4_RNG, SPORT_EQ_DPORT, SEQUENCE_EQ0, L4_FIN, L4_SYN, L4_RST, L4_PSH, L4_ACK, L4_URG, L4_1588_DOM, and L4_1588_VERSION) are automatically set to don't care.

TABLE 2-51: IS2 IP4_OTHER KEY

Field Name	Bit	Width	Description
Layer-3 Information			
IP4	46	1	Set if frame is IPv4 frame. IPv6 frames can also use IP4_OTHER entries when IP6_STD entries are disabled.
L3_FRAGMENT	47	1	Set if IP frame is fragmented (More Fragments flag = 1 or Fragments Offset > 0)
L3_FRAG_OFS_GT0	48	1	Set if IP frame is fragmented and if it is not the first fragment (Fragments Offset > 0).
L3_OPTIONS	49	1	Set if IPv4 frame contains options (IP len > 5). IP options are not skipped nor parsed, which implies that L3_PAYLOAD contains data from the IP options for IPv4 frames with IP options.
L3_TTL_GT0	50	1	Set if IP TTL is greater than 0.
L3_TOS	51	8	IP TOS field.
L3_IP4_DIP	59	32	IPv4 frames: Destination IPv4 address. IPv6 frames: Source IPv6 address, bits 63:32.
L3_IP4_SIP	91	32	IPv4 frames: Source IPv4 address. IPv6 frames: Source IPv6 address, bit 31:0.
DIP_EQ_SIP	123	1	Set if source IP address is equal to destination IP address. Full addresses are checked, also for IPv6.
L3_PROTO	124	8	IPv4: IP protocol. IPv6: next header.
L3_PAYLOAD	132	56	Bytes 0-6 after IP header.

TABLE 2-52: IS2 IP6_STD KEY

Field Name	Bit	Width	Description
Layer-3 Information			
L3_TTL_GT0	46	1	Set if IP HOPLIMIT is greater than 0.
L3_IP6_SIP	47	128	Frame's source IPv6 address
L3_PROTO	175	8	IPv6 next header.

TABLE 2-53: IS2 OAM KEY

Field Name	Bit	Width	Description
Layer-2 Information			
L2_DMAC	46	48	Frame's destination MAC address
L2_SMAC	94	48	Frame's source MAC address

TABLE 2-53: IS2 OAM KEY (CONTINUED)

Field Name	Bit	Width	Description
OAM_MEL_FLAGS	142	7	Encoding of MD level/MEG level (MEL). One bit for each level (lowest level encoded as zero) The following keys can be generated: MEL=0: 0x0000000 MEL=1: 0x0000001 MEL=2: 0x0000011 MEL=3: 0x0000111 MEL=4: 0x0001111 MEL=5: 0x0011111 MEL=6: 0x0111111 MEL=7: 0x1111111 Together with the mask, the following kinds of rules may be created: Exact match. Fx. MEL=2: 0x0000011 Below. Fx. MEL<=4: 0x000XXXX Above. Fx. MEL>=5: 0xXX11111 Between. Fx. 3<= MEL<=5: 0x00XX111, where ‘X’ means ‘don’t care’.
OAM_VER	149	5	Frame’s OAM version.
OAM_OPCODE	154	8	Frame’s OAM opcode.
OAM_FLAGS	162	8	Frame’s OAM flags.
OAM_MEPID	170	16	CCM frame’s OAM MEP ID.
OAM_C-CM_CNTS_EQ0	186	1	Flag indicating whether dual-ended loss measurement counters in CCM frames are used or not. The flag OAM_C-CM_CNTS_EQ0 is set if the counters TxFCf, RxFCb, and TxFCb are set to all zeros.
OAM_IS_Y1731	187	1	Set if frame’s EtherType = 0x8902

The following tables provide information about how the IS2 full keys are generated. The first table lists the common fields between the full keys.

TABLE 2-54: IS2 COMMON KEY FIELDS FOR FULL KEYS

Field Name	Bit	Width	Description
Lookup Information			
S2_TYPE_FULL	0	2	0: IP6_TCP_UDP entries. 1: IP6_OTHER entries. 2: Custom entries.
FIRST	2	1	Set for first lookup and cleared for second lookup.
Interface Information			
PAG	3	8	Policy association group. Action from VCAP IS1.
IGR_PORT_MASK	11	12	Ingress port mask. VCAP generated with one bit set in the mask corresponding to the ingress port.
Tagging and IP Source Guarding Information			
RESERVED	23	1	Reserved. Must be set to don’t care.
HOST_MATCH	24	1	The action from the SMAC_SIP4 or SMAC_SIP6 lookups. Used for IP source guarding.
L2_MC	25	1	Set if frame’s destination MAC address is a multicast address (bit 40 = 1).
L2_BC	26	1	Set if frame’s destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF).

TABLE 2-54: IS2 COMMON KEY FIELDS FOR FULL KEYS (CONTINUED)

Field Name	Bit	Width	Description
VLAN_TAGGED	27	1	Set if frame has one or more Q-tags. Independent of port VLAN awareness.
VID	28	12	Classified VID which is the result of the VLAN classification in basic classification and IS1.
DEI	40	1	Classified DEI, which is the final result of the VLAN classification in basic classification and IS1.
PCP	41	3	Classified PCP, which is the final result of the VLAN classification in basic classification and IS1.

TABLE 2-55: IS2 IP6_TCP_UDP KEY

Field Name	Bit	Width	Description
Layer-3 and Layer-4 Information			
L3_TTL_GT0	44	1	Set if IP HOPLIMIT is greater than 0.
L3_TOS	45	8	IP TOS field.
L3_IP6_DIP	53	128	Destination IPv6 address.
L3_IP6_SIP	181	128	Source IPv6 address.
DIP_EQ_SIP	309	1	Set if source IP address is equal to destination IP address.
TCP	310	1	Set if IP Proto = 6 (TCP).
L4_DPORT	311	16	TCP/UDP destination port.
L4_SPORT	327	16	TCP/UDP source port.
L4_RNG	343	8	Range mask with one bit per range. A bit is set, if the corresponding range is matched. Range types: SPORT, DPORT, SPORT or DPORT, VID, DSCP. Input to range checkers: SPORT, DPORT: From frame VID, DSCP: Classified result from IS1 See Section 2.8.5, Range Checkers.
SPORT_EQ_DPORT	351	1	Set if UDP or TCP source port equals UDP or TCP destination port.
SEQUENCE_EQ0	352	1	TCP: Set if TCP sequence number is 0. PTP over UDP: messageType bit 0.
L4_FIN	353	1	TCP: TCP flag FIN. PTP over UDP: messageType bit 1.
L4_SYN	354	1	TCP: TCP flag SYN. PTP over UDP: messageType bit 2.
L4_RST	355	1	TCP: TCP flag RST. PTP over UDP: messageType bit 3.
L4_PSH	356	1	TCP: TCP flag PSH. PTP over UDP: flagField bit 1 (twoStepFlag).
L4_ACK	357	1	TCP: TCP flag ACK. PTP over UDP: flagField bit 2 (unicastFlag).
L4_URG	358	1	TCP: TCP flag URG. PTP over UDP: flagField bit 7 (reserved).
L4_1588_DOM	359	8	PTP over UDP: domainNumber.
L4_1588_VERSION	367	4	PTP over UDP: version

TABLE 2-56: IS2 IP6_OTHER KEY

Field Name	Bit	Width	Description
Layer-3 Information			
L3_TTL_GT0	44	1	Set if IP HOPLIMIT is greater than 0.
L3_TOS	45	8	IP TOS field.
L3_IP6_DIP	53	128	Destination IPv6 address.
L3_IP6_SIP	181	128	Source IPv6 address.
DIP_EQ_SIP	309	1	Set if source IP address is equal to destination IP address.
L3_PROTO	310	8	IPv6 next header.
L3_PAYLOAD	318	56	Bytes 0-6 after IP header.

TABLE 2-57: IS2 CUSTOM KEY

Field Name	Bit	Width	Description
Custom Information			
CUSTOM_TYPE	44	1	0: CUSTOM_IP - Custom data extracted from byte 34 (corresponds to the first byte after the IPv4 header if the frame is IPv4). 1: CUSTOM_SMAC - Custom data extracted from byte 12 (first byte after SMAC). Note that up to two VLAN tags are removed before extracting the data.
CUSTOM	45	320	Custom extracted data.

2.8.3.1.1 IS2 Action Encoding

The VCAP generates an action vector from each of the two IS2 lookups for each frame.

The first IS2 lookup returns a default action vector per ingress port when no entries are matched, and the second IS2 lookup returns a common default action vector when no entries are matched. There are no difference between an actionvector from a match and a default action vector.

The following table lists the available fields for the action vector.

TABLE 2-58: IS2 ACTION FIELDS

Action Field	Bit	Width	Description
HIT_ME_ONCE	0	1	Setting this bit to 1 causes the first frame that hits this action where the HIT_CNT counter is zero to be copied to the CPU extraction queue specified in CPU_QU_NUM. The HIT_CNT counter is then incremented and any frames that hit this action later are not copied to the CPU. To re-enable the HIT_ME_ONCE functionality, the HIT_CNT counter must be cleared.
CPU_COPY_ENA	1	1	Setting this bit to 1 causes all frames that hit this action to be copied to the CPU extraction queue specified in CPU_QU_NUM.
CPU_QU_NUM	2	3	Determines the CPU extraction queue that is used when a frame is copied to the CPU due to a HIT_ME_ONCE or CPU_COPY_ENA action.

TABLE 2-58: IS2 ACTION FIELDS (CONTINUED)

Action Field	Bit	Width	Description
MASK_MODE	5	2	Controls how PORT_MASK is applied. 0: No action from PORT_MASK. 1: Permit/deny (PORT_MASK AND'ed with destination set). 2: Policy forwarding (DMAC lookup replaced with PORT_MASK). 3: Redirect (Previous forwarding decisions (classifier, IS1, ISDX, SRC, AGGR, VLAN, and DMAC lookup) are replaced with PORT_MASK). The CPU port is not touched by MASK_MODE.
MIRROR_ENA	7	1	Setting this bit to 1 causes frames to be mirrored to the mirror target port (ANA::MIRRPORPORTS).
LRN_DIS	8	1	Setting this bit to 1 disables learning of frames hitting this action.
POLICE_ENA	9	1	Setting this bit to 1 causes frames that hit this action to be policed by the ACL policer specified in POLICE_IDX. Only applies to the first lookup.
POLICE_IDX	10	9	Selects VCAP policer used when policing frames (POLICE_ENA).
POLICE_VCAP_ONLY	19	1	Disable policing from QoS, and port policers. Only the VCAP policer selected by POLICE_IDX is active. Only applies to the second lookup.
PORT_MASK	20	11	Port mask applied to the forwarding decision based on MASK_MODE.
REW_OP	31	9	Rewriter operation command. The following functions are supported: No operation: REW_OP[3:0] = 0. No operation. Special Rewrite: REW_OP[3:0]= 8. Swap the MAC addresses and clear bit 40 in the new SMAC when transmitting the frame. One-step PTP: REW_OP[2:0] = 2. The frame's residence time is added to the correction field in the PTP frame. The following sub-commands can be encoded: REW_OP[7]: Enables Add/subtract mode. REW_OP[5]: Set if egress delay must be added to correction field at egress. REW_OP[4:3]: Configures if ingress delay must be subtracted from the frame's Rx time stamp. This is not applicable for ingress ports in backplane mode. Bits 4:3 = 0: No delay adjustments. Bits 4:3 = 1: Subtract delay ANA:PORT:PTP_DLY1_CFG. Bits 4:3 = 2: Subtract delay ANA:PORT:PTP_DLY2_CFG. Two-step PTP: REW_OP[2:0] = 3. The frame's departure time stamp is saved in the time stamp FIFO queue at egress. REW_OP[8:3] holds the PTP time stamp identifier used by the CPU. Origin PTP: REW_OP[2:0] = 9. The time of day at the frame's departure time is written into the origin time stamp field in the PTP frame. Unspecified bits must be set to 0.

TABLE 2-58: IS2 ACTION FIELDS (CONTINUED)

Action Field	Bit	Width	Description
SMAC_RE-PLACE_ENA	40	1	If set, frame's source MAC address is replaced with MAC address defined per egress port (SYS::REW_MAC_LOW_CFG, SYS::REW_MAC_HIGH_CFG)
RESERVED	41	2	Reserved. Must be set to 0.
ACL_ID	43	6	Logical ID for the entry. This ID is extracted together with the frame in the CPU extraction header. Only applicable to actions with CPU_COPY_ENA or HIT_ME_ONCE set.
HIT_CNT	49	32	A statistics counter that is incremented by one each time the given action is hit.

The two action vectors from the first and second lookups are combined into one action vector, which is applied in the analyzer. For more information, see Section 2.9.3, Forwarding Engine. The actions are combined as follows:

- HIT_ME_ONCE, CPU_COPY_ENA, CPU_QU_NUM:
  If any of the two action vectors have HIT_ME_ONCE or CPU_COPY_ENA set, CPU_COPY_ENA is forwarded to the analyzer. The settings in the action vector from second lookup takes precedence with respect to the CPU extraction queue number.
- MIRROR_ENA:
  If any of the two action vectors have MIRROR_ENA set, MIRROR_ENA is forwarded to the analyzer.
- LRN_DIS:
  If any of the two action vectors have LRN_DIS set, LRN_DIS is forwarded to the analyzer.
- REW_OP:
  The settings in the action vector from the second lookup takes precedence unless the second lookup returns REW_OP[3:0] = 0.
- ACL_ID:
  If both lookups in IS2 hit an entry with CPU_COPY_ENA set, then the resulting value is the addition of the two ACL_ID values. This allow for unique identification of both rules in IS2 through ACL_ID if each lookup manipulates its own part of the ACL_ID. For instance, first lookup returns ACL_ID = 0:15, and second lookup returns ACL_ID = 0, 16, 32, or 48.
- POLICE_ENA, POLICE_IDX, POLICE_VCAP_ONLY:
  Only applies to actions from the first lookup.

The following table lists the combinations for `MASK_MODE` and `PORT_MASK` when combining actions from the first and second lookups.

TABLE 2-59: MASK_MODE AND PORT_MASK COMBINATIONS

	Second Lookup			
First Lookup	**No action**	**Permit/deny**	**Policy**	**Redirect**
No action	No action	Permit P[1] = P2[2]	Policy P = P2	Redirect P = P2
Permit/deny	Permit P = P1[3]	Permit P = P1 and P2	Policy P = P1 and P2	Redirect P = P2
Policy	Policy P = P1	Policy P = P1 and P2	Policy P = P1 and P2	Redirect P = P2
Redirect	Redirect P = P1	Redirect P = P1 and P2	Redirect P = P1 and P2	Redirect P = P2

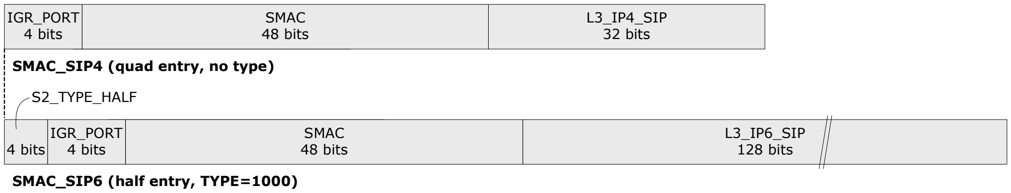
[1] P: Resulting `PORT_MASK` to analyzer.
[2] P2: PORT_MASK from second match.
[3] *P1:* PORT_MASK *from first match.*

Policy forwarding for frames matching an IPv4 and IPv6 multicast entry in the MAC table is not possible. Policy forwarding is handled as a permit/deny action for such frames.

2.8.3.1.2 SMAC_SIP4 and SMAC_SIP6 Entry Key Encoding

The following illustration shows which entry fields are available for SMAC_SIP4 and SMAC_SIP6 keys in IS2.

FIGURE 2-21: SMAC_SIP ENTRY TYPE OVERVIEW



VCAP IS2 can hold the following number of SMAC_SIP entries.

- SMAC_SIP6: Up to 128 entries.
- SMAC_SIP4: Up to 256 entries.

All IPv6 frames are subject to a `SMAC_SIP6` lookup in IS2. The following table lists the `SMAC_SIP6` key.

TABLE 2-60: SMAC_SIP6 KEY

Field Name	Bit	Width	Description
Lookup Information			
S2_TYPE_HALF	0	4	8: SMAC_SIP4.
Interface Information			
IGR_PORT	4	4	The port number where the frame was received (0-11).
Layer-2 Information			
L2_SMAC	8	48	Frame’s source MAC address.
Layer-3 Information			
L3_IP6_SIP	56	128	Frame’s source IPv6 address.

All IPv4 frames are subject to a SMAC_SIP4 lookup in IS2. The following table lists the SMAC_SIP4 key.

TABLE 2-61: SMAC_SIP4 KEY

Field Name	Bit	Width	Description
Interface Information			
IGR_PORT	0	4	The port number where the frame was received (0-11).
Layer-2 Information			
L2_SMAC	4	48	Frame’s source MAC address.
Layer-3 Information			
L3_IP4_SIP	52	32	Frame’s source IPv4 address.

2.8.3.1.3 SMAC_SIP4 and SMAC_SIP6 Action Encoding

The VCAP generates an action vector from the SMAC_SIP4 or SMAC_SIP6 lookup if there is a match. There is no default action vector if no match.

The following table lists the available fields for the action vector from both the SMAC_SIP4 and SMAC_SIP6 lookups.

TABLE 2-62: SMAC_SIP4 AND SMAC_SIP6 ACTION FIELDS

Action Field	Bit	Width	Description
CPU_COPY_ENA	0	1	Setting this bit to 1 causes all frames that hit this action to be copied to the CPU extraction queue specified in CPU_QU_NUM.

TABLE 2-62: SMAC_SIP4 AND SMAC_SIP6 ACTION FIELDS (CONTINUED)

Action Field	Bit	Width	Description
CPU_QU_NUM	1	3	Determines the CPU extraction queue that is used when a frame is copied to the CPU due to a HIT_ME_ONCE or CPU_COPY_ENA action.
FWD_KILL_ENA	4	1	Setting this bit to 1 denies forwarding of the frame forwarding to any front port. The frame can still be copied to the CPU by other actions.
HOST_MATCH	5	1	Used for IP source guarding. If set, it signals that the host is a valid (for instance a valid combination of source MAC address and source IP address). HOST_MATCH is input to the IS2 keys.
HIT_STICKY	6	1	If set, a frame has matched against the associated entry.

The HOST_MATCH flag is used as input into the IS2 keys. This enables further handling of frames either matching or not matching a host pair in the SMAC_SIP table.

2.8.4 VCAP ES0

This section provides information about the ES0 key and associated actions.

2.8.4.1 ES0 Entry Key Encoding

All frames are subject to one ES0 lookup per destination port. The key in ES0 is generated based on the frame’s VLAN classification.

VCAP ES0 can hold 256 ES0 entries.

The following table lists the ES0 key.

TABLE 2-63: ES0 KEY

Field Name	Bit	Width	Description
Interface Information			
EGR_PORT	0	4	The port number where the frame is transmitted (0-11).
IGR_PORT	4	4	The port number where the frame was received (0-11).
RESERVED	8	2	Reserved. Must be set to don’t care.
L2_MC	10	1	Set if frame’s destination MAC address is a multicast address (bit 40 = 1).
L2_BC	11	1	Set if frame’s destination MAC address is the broadcast address (FF-FF-FF-FF-FF-FF).
Tagging Information			
VID	12	12	Classified VID that is the result of the VLAN classification in basic classification and IS1.
DP	24	1	Frame’s drop precedence (DP) level after policing.
PCP	25	3	Classified PCP that is the final result of the VLAN classification in basic classification and IS1.

2.8.4.2 ES0 Action Encoding

The VCAP generates one action vector from the ES0 lookup. The lookup returns a default action vector per egress port when no entries are matched. There are no difference between an action vector from a match and a default action vector.

The following table lists the available action fields for ES0. For more information about how the actions are applied to the VLAN manipulations, see Section 2.13.1, VLAN Editing.

TABLE 2-64: ES0 ACTION FIELDS

Action Field	Bit	Width	Description
PUSH_OUTER_TAG	0	2	Controls outer tagging. 0: No ES0 tag A: Port tag is allowed if enabled on port. 1: ES0 tag A: Push ES0 tag A. No port tag. 2: Force port tag: Always push port tag. No ES0 tag A. 3: Force untag: Never push port tag or ES0 tag A.
PUSH_INNER_TAG	2	1	Controls inner tagging. 0: Do not push ES0 tag B as inner tag. 1: Push ES0 tag B as inner tag.
TAG_A_TPID_SEL	3	2	Selects TPID for ES0 tag A: 0: 0x8100. 1: 0x88A8. 2: Custom (REW:PORT:PORT_VLAN_CFG.PORT_TPID). 3: If IFH.TAG_TYPE = 0 then 0x8100 else custom.
TAG_A_VID_SEL	5	1	Selects VID for ES0 tag A. 0: Classified VID + VID_A_VAL. 1: VID_A_VAL.
TAG_A_PCP_SEL	6	2	Selects PCP for ES0 tag A. 0: Classified PCP. 1: PCP_A_VAL. 2: DP and QoS mapped to PCP (per port table). 3: QoS class.
TAG_A_DEI_SEL	8	2	Selects PCP for ES0 tag A. 0: Classified DEI. 1: DEI_A_VAL. 2: DP and QoS mapped to PCP (per port table). 3: DP.
TAG_B_TPID_SEL	10	2	Selects TPID for ES0 tag B. 0: 0x8100. 1: 0x88A8. 2: Custom (REW:PORT:PORT_VLAN_CFG.PORT_TPID). 3: If IFH.TAG_TYPE = 0 then 0x8100 else custom.
TAG_B_VID_SEL	12	1	Selects VID for ES0 tag B. 0: Classified VID + VID_B_VAL. 1: VID_B_VAL.
TAG_B_PCP_SEL	13	2	Selects PCP for ES0 tag B. 0: Classified PCP. 1: PCP_B_VAL. 2: DP and QoS mapped to PCP (per port table). 3: QoS class.
TAG_B_DEI_SEL	15	2	Selects PCP for ES0 tag B. 0: Classified DEI. 1: DEI_B_VAL. 2: DP and QoS mapped to PCP (per port table). 3: DP.
VID_A_VAL	17	12	VID used in ES0 tag A. See TAG_A_VID_SEL.
PCP_A_VAL	29	3	PCP used in ES0 tag A. See TAG_A_PCP_SEL.
DEI_A_VAL	32	1	DEI used in ES0 tag A. See TAG_A_DEI_SEL.
VID_B_VAL	33	12	VID used in ES0 tag B. See TAG_B_VID_SEL.
PCP_B_VAL	45	3	PCP used in ES0 tag B. See TAG_B_PCP_SEL.

TABLE 2-64: ES0 ACTION FIELDS (CONTINUED)

Action Field	Bit	Width	Description
DEI_B_VAL	48	1	DEI used in ES0 tag B. See TAG_B_DEI_SEL.
RESERVED	49	24	Reserved. Must be set to 0.
HIT_STICKY	73	1	If set, a frame has matched the associated entry.

2.8.5 RANGE CHECKERS

The following table lists the registers associated with configuring range checkers.

TABLE 2-65: RANGE CHECKER CONFIGURATION

Register	Description	Replication
ANA::VCAP_RNG_TYPE_CFG	Configuration of the range checker types.	None
ANA::VCAP_RNG_VAL_CFG	Configuration of range start and end points.	None

All IS1 entries, together with the IP4_TCP_UDP and IP6_TCP_UDP entries in IS2, contain eight range checker flags (L4_RNG), which are matched against an 8-bit range key. The range key is generated for each frame based on the extracted frame data and the configuration in ANA::VCAP_RNG_TYPE_CFG and ANA::VCAP_RNG_VAL_CFG. Each of the eight range checkers can be configured to one of the following range types:

- TCP/UDP destination port range
  Input to the range is the frame’s TCP/UDP destination port number.
- TCP/UDP source port range
  Input to the range is the frame’s TCP/UDP source port number.
- TCP/UDP source and destination ports range. Range is matched if either source or destination port is within range.
  Input to the range are the frame’s TCP/UDP source and destination port numbers.
- VID range
  IS1: Input to the range is the frame’s VID or the port VID if the frame is untagged.
  IS2: Input to the range is the classified VID.
- DSCP range
  IS1: Input to the range is the translated DSCP value from basic classification.
  IS2: Input to the range is the classified DSCP value.

For IS2, the range key is only applicable to TCP/UDP frames. For IS1, the range key is generated for any frame types. Specific range types not applicable to a certain frame type (for example, TCP/UDP port ranges for IPv4 Other frames) must be set to don’t care in entries the frame type can match.

Range start points and range end points are configured in ANA::VCAP_RNG_VAL_CFG.

2.8.6 VCAP CONFIGURATION

This section provides information about how the VCAPs (IS1, IS2, and ES0) are configured.

Each VCAP implements its own set of the registers listed in the following two tables.

Entries in a VCAP are accessed indirectly through an entry and action cache. The cache is accessible using the VCAP configuration registers listed in following table. As shown in the following illustration, an entry in the VCAP consists of a TCAM entry and an associated action and counter entry.

The following table lists the registers associated with VCAP configuration.

TABLE 2-66: VCAP CONFIGURATION REGISTERS

Register	Description	Replication
VCAP_UPDATE_CTRL	General configuration register	None
VCAP_MV_CFG	Move configuration	None
VCAP_ENTRY_DAT	Entry data cache	64
VCAP_MASK_DAT	Entry mask cache	64
VCAP_ACTION_DAT	Action data cache	64

TABLE 2-66: VCAP CONFIGURATION REGISTERS (CONTINUED)

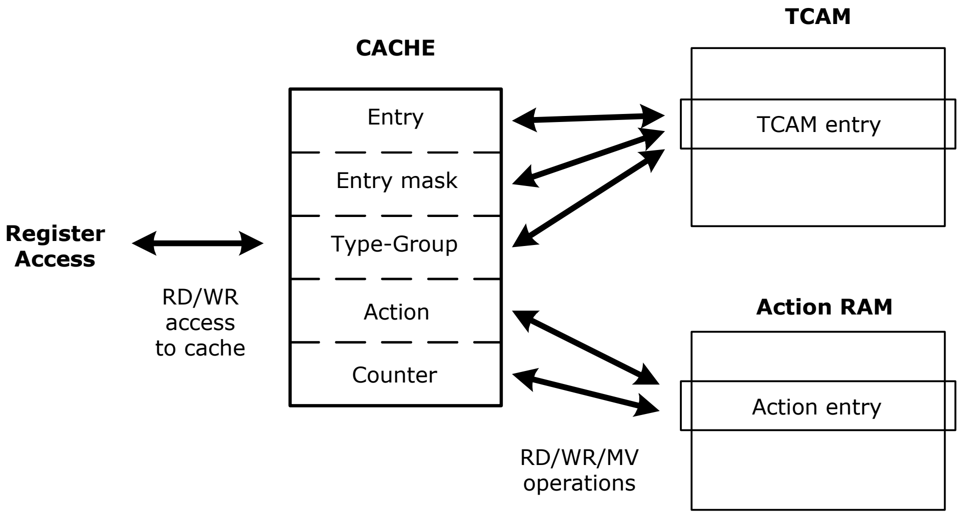
Register	Description	Replication
VCAP_CNT_DAT	Counter data cache	32
VCAP_TG_DAT	Type-Group cache	None
VCAP_STICKY	Sticky-bit indications	None

Each VCAP has defined various constants and are accessed using the registers listed in the following table.

TABLE 2-67: VCAP CONSTANTS

Register	Description	Replication
ENTRY_WIDTH	Width of entry field	None
ENTRY_CNT	Number of entries	None
ENTRY_SWCNT	Number of subwords	None
ENTRY_TG_WIDTH	Width of type-group field	None
ACTION_DEF_CNT	Number of default actions	None
ACTION_WIDTH	Width of action field	None
CNT_WIDTH	Width of counter field	None

FIGURE 2-22: VCAP CONFIGURATION OVERVIEW



A TCAM entry consists of entry data, an entry mask, and a type-group value. The type-group value is used internally to differentiate between VCAP lookups of different subword sizes. Each TCAM entry has an associated action entry. In addition, the action RAM has an entry for each of the default actions in the VCAP. The entries in the action RAM consists of action data and a counter value.

For a write access, the TCAM and action entry must be written to the cache and then copied from the cache to the TCAM/RAM. For a read access, the TCAM and action entry must first be retrieved from the TCAM/RAM before being read from the cache. When a read or write operation is initiated, it is possible to individually select if the operation should be applied to the TCAM and/or action RAM. When data is moved between the cache and the TCAM/RAM, it is always the entire entry that is moved. For VCAPs with several subwords per entry, this must be taken into account if only a single subword of a TCAM entry should be updated. To modify a single subword, the entire TCAM entry must be read, then the subword must be modified in the cache, and finally the entry must be written back to the TCAM.

The cache can hold only one VCAP entry (TCAM and action entry) at a time. After the TCAM and action entry are written to the cache, the cache must be copied to the TCAM and RAM before new entries can be written to the cache.

The following table lists the different parameters for the four VCAPs available in the device. The parameters are needed to format the data to be written to the cache. The parameters can also be read in the registers listed in Table 2-67.

TABLE 2-68: VCAP PARAMETERS

VCAP	Entry Width	Number of Entries	Action Width	Number of Default Actions	Counter Width	Subwords	Type-Group Width
IS1	384	64	312	1	4x1 (sticky)	4	2
IS2	384	64	99	13	4x32	4	3
ES0	29	256	72	11	1 (sticky)	1	1

2.8.6.1 Creating a VCAP Entry in the Cache

Before a VCAP entry can be created in the TCAM and RAM, the entry must be created in the cache. The cache is accessed through the following 32-bit registers.

- VCAP_ENTRY_DAT
- VCAP_MASK_DAT
- VCAP_ACTION_DAT
- VCAP_CNT_DAT
- VCAP_TG_DAT

Each of the cache registers is replicated 64 times; however, only the bits used by the VCAP are mapped to physical registers. For example, for VCAP IS1, only the lowest 384 bits of VCAP_ENTRY_DAT and VCAP_MASK_DAT is mapped to physical registers. As mentioned previously, a VCAP entry consists of a TCAM entry and an action entry.

The TCAM entry consists of entry data, mask data, a type value, and a type-group value. The entry data prefixed with the type value is written to VCAP_ENTRY_DATA. The mask data is written to VCAP_MASK_DATA, and the type-group value is written to VCAP_TG_DAT. The type and type-group values are used internally in the VCAP to distinguish between the different entry types. The following table lists the type and type-group value for each of the entry types.

TABLE 2-69: ENTRY, TYPE, AND TYPE-GROUP PARAMETERS

VCAP	Entry Type	Entry Width	Subwords	Type Value [width in ()]	Type-Group Value [width in ()]
IS1	S1_NORMAL	180	2	0 (1)	2 (2)
IS1	S1_5TUPLE_IP4	187	2	1 (1)	2 (2)
IS1	S1_NORMAL_IP6	374	1	0 (2)	1 (2)
IS1	S1_7TUPLE	373	1	1 (2)	1 (2)
IS1	S1_5TUPLE_IP6	374	1	2 (2)	1 (2)
IS1	S1_DBL_VID	93	4	Not used (0)	3 (2)
IS2	MAC_ETYPE	181	2	0 (4)	2 (2)
IS2	MAC_LCC	178	2	1 (4)	2 (2)
IS2	MAC_SNAP	178	2	2 (4)	2 (2)
IS2	ARP	163	2	3 (4)	2 (2)
IS2	IP4_TCP_UCP	181	2	4 (4)	2 (2)
IS2	IP4_OTHER	184	2	5 (4)	2 (2)
IS2	IP6_STD	179	2	6 (4)	2 (2)
IS2	OAM	184	2	7 (4)	2 (2)
IS2	IP6_TCP_UDP	369	1	0 (2)	1 (2)
IS2	IP6_OTHER	372	1	1 (2)	1 (2)
IS2	CUSTOM	363	1	2 (2)	1 (2)
IS2	SMAC_SIP4	84	4	Not used (0)	3 (2)
IS2	SMAC_SIP6	180	2	8 (4)	2 (2)

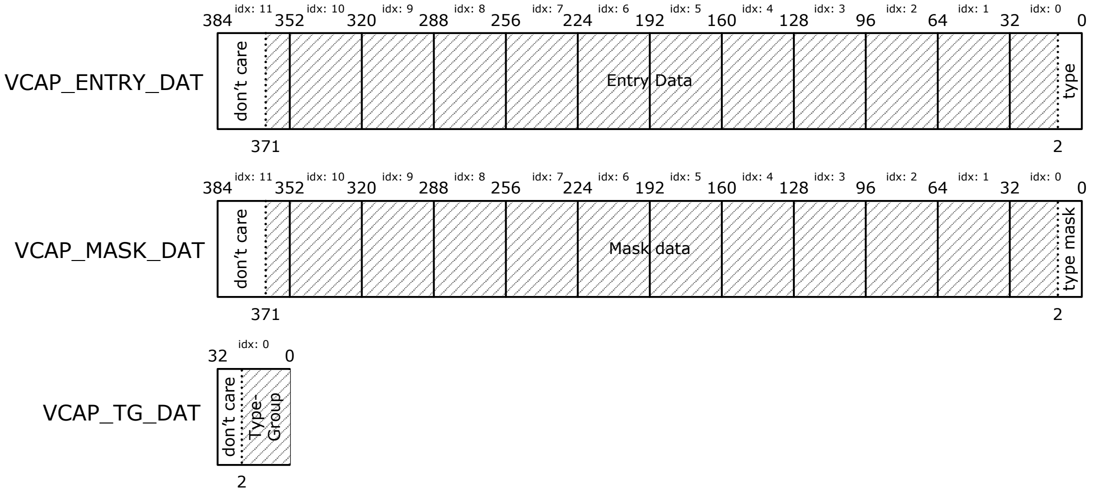
TABLE 2-69: ENTRY, TYPE, AND TYPE-GROUP PARAMETERS (CONTINUED)

VCAP	Entry Type	Entry Width	Subwords	Type Value [width in ()]	Type-Group Value [width in ()]
ES0	VID	28	1	Not used (0)	1 (1)

Note that the type value is not used for all entry types. If the type value is not used for an entry type, write the entry data from bit 0 of VCAP_ENTRY_DAT.

As an example of how a TCAM entry is laid out in the cache register, the following illustration shows a TCAM entry of the IP6_TCP_UDP entry type for the VCAP IS2.

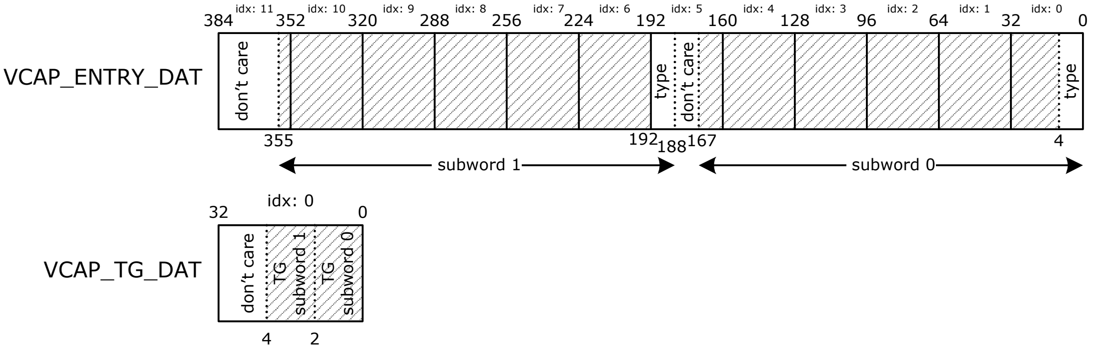
FIGURE 2-23: ENTRY LAYOUT IN REGISTER EXAMPLE



Generally, the type value must never be masked. However, by masking the type bits a lookup in the VCAP is able to match several different entry types. For example, the IS2 entry types MAC_ETYPE and MAC_LLC have the binary type values 0000 and 0001, respectively. By masking bit 0, a lookup is able to match both entry types.

The entry type used in the preceding example only has one subword per entry in the TCAM. Creating a TCAM entry with an entry type that has several subwords per TCAM entry is a little more complicated. As shown in the following example, the ARP entry type of the VCAP IS2 is used. The ARP entry type has two subwords per TCAM entry. From Table 2-69, it can be seen that the ARP entry type has a width of 163 bits per subword. A row in the IS2 TCAM is 384 bits wide. For more information, see Table 2-68. Each subword is assigned to half a TCAM row; that is, subword 0 is assigned to bits 0-187 and subword 1 is assigned to bits 188-375. Because the ARP entry only is 167 bits wide, there are 21 unused bits for each subword, as shown in the following illustration. The layout for VCAP_MASK_DAT is similar to VCAP_ENTRY_DAT. In addition, a type-group value is associated to each subword. The type-group values are laid out back-to-back in VCAP_TG_DAT as shown in the following illustration.

FIGURE 2-24: ENTRY LAYOUT IN REGISTER USING SUBWORDS EXAMPLE



To invalidate an entry in the TCAM (so a lookup never matches the entry), set the type-group for the entry to 0. If there are more subwords in the entry, each subword can be individually invalidated by setting its corresponding type-group value to 0.

The action entry is written to VCAP_ACTION_DAT. Similar to an entry data, an action entry also has a prefixed type value. The following table lists the parameters for the different action types available in VCAPs.

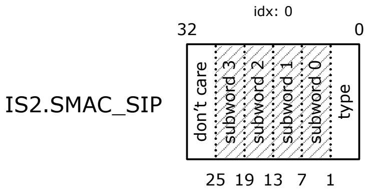
TABLE 2-70: ACTION AND TYPE FIELD PARAMETERS

VCAP	Action Type	Action Width	Subwords	Type Value [width in ()]
IS1	S1	80	4	Not used (0)
IS2	BASE_TYPE	51	2	0 (1)
IS2	SMAC_SIP	6	4	1 (1)
ES0	VID	91	1	Not used (0)

An action that is associated with an entry type with several subwords per entry has an equal number of subwords. For actions with several subwords, the subwords are simply concatenated together.

The following illustration shows the action layout in the VCAP_ACTION_DAT register for an SMAC_SIP action entry. The SMAC_SIP has four subwords per row.

FIGURE 2-25: ACTION LAYOUT IN REGISTER EXAMPLE



The counter value associated to the action is written to VCAP_CNT_DAT. VCAP_CNT_DAT contains a counter value for each subword in the TCAM entry. For action entries, the counter values for each subword are simply concatenated together.

2.8.6.2 Copying Entries Between Cache and TCAM/RAM

When an entry and associated action is created in the cache, the data in the cache must be copied to a given address in the TCAM and RAM using the VCAP_UPDATE_CTRL register. Use the following procedure.

1. Set VCAP_UPDATE_CTRL.UPDATE_CMD to copy from cache to TCAM/RAM.
2. Set the address for the entry in VCAP_UPDATE_CTRL.UPDATE_ADDR.
3. Set VCAP_UPDATE_CTRL.UPDATE_SHOT to initiate the copy operation. The bit is cleared by hardware when the operation is finished.

Initiating another operation before the UPDATE_SHOT field is cleared is not allowed. The delay between setting the UPDATE_SHOT field and the clearing of that field depends on the type of operation and the traffic load on the VCAP.

By setting the fields UPDATE_ENTRY_DIS, UPDATE_ACTION_DIS, and/or UPDATE_CNT_DIS in the VCAP_UPDATE_CTRL register the writing of the TCAM, action, and/or the counter entry can be disabled.

Copying a VCAP entry from the TCAM/RAM to the cache is done in a similar fashion by setting VCAP_UPDATE_CTRL.UPDATE_CMD to copy from TCAM/RAM to the cache. Note that due to internal mapping of the entry data and mask data, the values that are read back from the TCAM cannot always match with the values that were originally written to the TCAM. The internal mapping that happens is listed in the following table. There are differences, because a masked 1 is read back as a masked 0, which is functionally the same.

TABLE 2-71: INTERNAL MAPPING OF ENTRY AND MASK

Written Entry	Written Mask	Description	Read Entry	Read Mask
0	0	Match-0	0	0
0	1	Match-Any	0	1

TABLE 2-71: INTERNAL MAPPING OF ENTRY AND MASK (CONTINUED)

Written Entry	Written Mask	Description	Read Entry	Read Mask
1	0	Match-1	1	0
1	1	Match-Any	0	1

If an entry match is not found during a lookup for a given frame, a default action is selected by the VCAP. Default actions and counter values are copied between the cache and the action RAM similar to a regular VCAP entry. The default actions are stored in the RAM right below the last regular action entry; for example, VCAP IS2 has 256 regular entries, so the first default action in VCAP IS2 is stored at address 256, the second at address 257, and so on. For more information about the number of regular VCAP entries in each VCAP, see Table 2-68. When a default action is copied from the cache to the RAM, VCAP_UPDATE_CTRL.UPDATE_ENTRY_DIS must be set to disable the update of the TCAM. If updating of the TCAM is not disabled, the operation may overwrite entries in the TCAM.

The cache can be cleared by setting VCAP_UPDATE_CTRL.CLEAR_CACHE. This sets all replications of VCAP_ENTRY_DAT, VCAP_MASK_DAT, VCAP_ACTION_DAT, VCAP_CNT_DAT, and VCAP_TG_DAT to zeros. The CLEAR_CACHE field is automatically cleared by hardware when the cache is cleared.

2.8.7 ADVANCED VCAP OPERATIONS

The VCAP supports a number of advanced operations that allow easy moving and removal of entries and actions during frame traffic.

2.8.7.1 Moving a Block of Entries and Actions

A number of entries and actions can be moved up or down by several positions in the TCAM and RAM. This is done using the VCAP_UPDATE_CTRL and VCAP_MV_CFG registers.

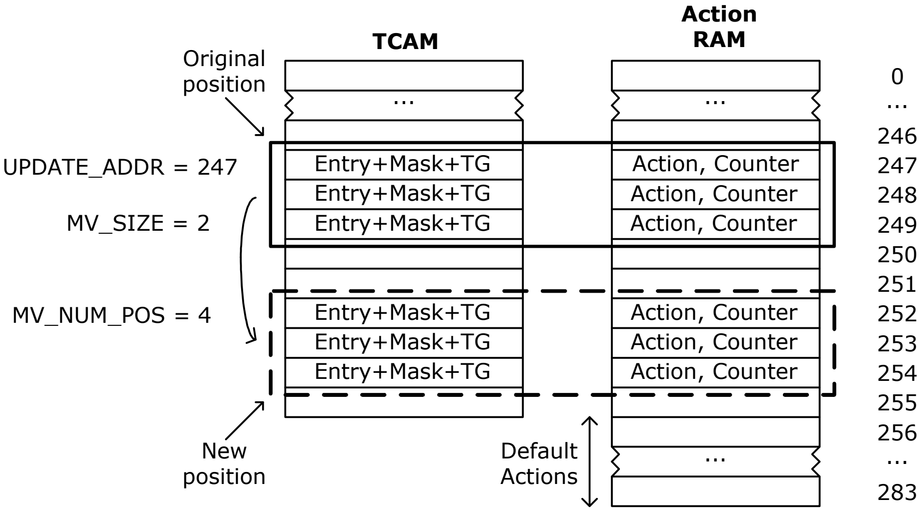
A Move operation is performed by:

- Setting VCAP_UPDATE_CTRL.UPDATE_ADDR equal to the address of the entry with the lowest address, among the entries that must be moved.
- Setting VCAP_MV_CFG.MV_SIZE to the number of entries that must be moved; n + 1 entries are moved.
- Setting VCAP_MV_CFG.MV_NUM_POS to the number of positions the entries must be moved. The entries are moved n positions up or down.
- Setting UPDATE_ENTRY_DIS, UPDATE_ACTION_DIS, and/or UPDATE_CNT_DIS to only move some parts of the VCAP entry.
- Setting VCAP_UPDATE_CTRL.UPDATE_CMD to move up (decreasing addresses) or move down (increasing addresses).
- Initiating the Move operation by setting VCAP_UPDATE_CTRL.VCAP_UPDATE_SHOT.

A new command must not be setup until after the VCAP_UPDATE_CTRL.VCAP_UPDATE_SHOT field has automatically cleared. Also note that the cache is used by the VCAP while a Move operation is being performed. As a result, any value in cache prior to a Move operation is lost, and a write is not permitted to the cache while a Move operation is performed.

The following illustration shows an example of a Move operation.

FIGURE 2-26: MOVE DOWN OPERATION EXAMPLE



A Move operation can be performed hitlessly during frame traffic, that is, all entries and actions are still available during a Move operation, and all hits are counted by the action hit counters. The TCAM entries at the original positions are invalidated after the Move operation is complete.

During heavy frame traffic, it can take some time for a large move operation to complete, because the moving of individual rows are restarted each time a lookup is performed. If it is not important that the hit counters are accurately updated while the move operation is processed, VCAP_UPDATE_CTRL.MV_TRAFFIC_IGN can be set. This prevents the VCAP from restarting moves and consequently, decreases the time it takes for the move operation to complete. It may, however, lead to inaccurate hit counter values. Note that even if MV_TRAFFIC_IGN is set, the VCAP still processes all lookups correctly.

Default actions can also be moved, however, VCAP_UPDATE_CTRL.UPDATE_ENTRY_DIS must be set.

If a row is moved to a negative address (above address 0), the row is effectively deleted. If a block is partly moved above address 0, the block is also only partially deleted. In other words, the rows that are effectively moved to an address below 0 are not deleted. If one or more rows are deleted during a move operation, the sticky bit VCAP_-STICKY.VCAP_ROW_DELETED_STICKY is set.

2.8.7.2 Initializing A Block of Entries

A block of entries can be set to the value of the cache in a single operation. For example, it can be used to initialize all TCAM, action, and counter entries to a specific value. The block of entries to initialize can also include the default action and counter entries.

To perform an initialization operation:

- Set VCAP_UPDATE_CTRL.UPDATE_ADDR equal to the address of the entry with the lowest address, among the entries that should be written.
- Set VCAP_MV_CFG.MV_SIZE to the number of entries that must be included in the initialization operation: n + 1 entries are included.
- Set UPDATE_ENTRY_DIS, UPDATE_ACTION_DIS, and/or UPDATE_CNT_DIS to select if the TCAM, action RAM, and/or the counter RAM should be excluded from the initialization operation.
- Set VCAP_UPDATE_CTRL.UPDATE_CMD to the initialization operation.
- Start the initialization operation by setting VCAP_UPDATE_CTRL.VCAP_UPDATE_SHOT.

A new command must not be set up until after the VCAP_UPDATE_CTRL.VCAP_UPDATE_SHOT field is automatically cleared neither must the cache be written to before VCAP_UPDATE_SHOT is cleared.

2.9 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
- VLAN table
- Forwarding Engine

The MAC 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 2.8, 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 11 bits, 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.

2.9.1 MAC TABLE

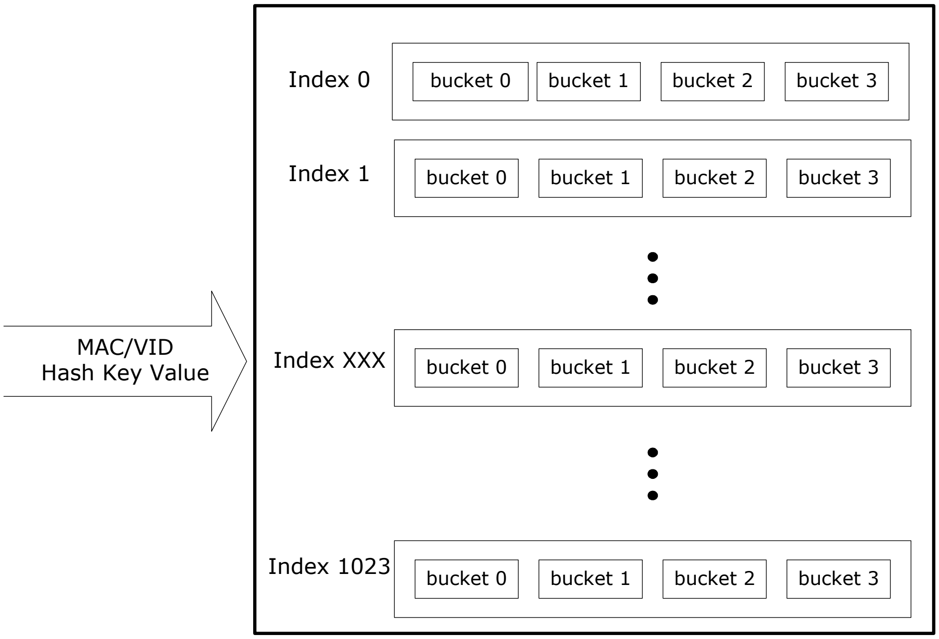
This section provides information about the MAC table block in the analyzer. The following table lists the registers associated with MAC table access.

TABLE 2-72: MAC TABLE ACCESS

Register	Description	Replication
MACHDATA	MAC address and VID when accessing the MAC table.	None
MACLDATA	MAC address when accessing the MAC table.	None
MACTINDX	Direct address into the MAC table for direct read and write.	None
MACACCESS	Flags and command when accessing the MAC table.	None
MACTOPTIONS	Flags when accessing the MAC table.	None
AUTOAGE	Age scan period.	None
AGENCTRL	Controls the default values for new entries in MAC table.	None
ENTRYLIM	Controls limits on number of learned entries per port.	Per port
LEARNDISC	Counts the number of MAC table entries not learned due lack of storage in the MAC table.	None

The analyzer contains a MAC table with 4096 entries containing information about stations learned by the device. The table is organized as a hash table with four buckets and 1024 rows. Each row is indexed by an 10-bit hash value, which is calculated based on the station's (MAC, VID) pair, as shown in the following illustration.

FIGURE 2-27: MAC TABLE ORGANIZATION



The following table lists the fields for each entry in the MAC table.

TABLE 2-73: 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).
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 2.9.1.2, 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. **Note:** 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.

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).

2.9.1.1 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 prepended with 0.
- 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 2.9.3.5, SMAC Analysis.

2.9.1.2 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 2.9.1.1, Hardware-Based Learning.

2.9.1.3 CPU Commands

The following table 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 2-74: 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.

TABLE 2-74: MAC TABLE COMMANDS (CONTINUED)

Command	Purpose	Use
READ	Read entry pointed to by (row, column).	Configure row (0-1023) 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.
IDLE	Indicate that MAC table is ready for new command.	No preload required.

2.9.1.4 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.

2.9.1.5 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 the following table.

TABLE 2-75: IPV4 MULTICAST DESTINATION MASK

Destination Ports	Record Bit Field
Ports 10-0	MAC[34-24]

Example: All IPv4 multicast frames in VLAN 12 with MAC 01005E112233 are to be forwarded to ports 3, 8, and 9. This is done by inserting the following entry in the MAC table entry:

VALID = 1
VID = 12
MAC = 0x000308112233
ENTRY_TYPE = 2
DEST_IDX = 0

2.9.1.6 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 the following table.

TABLE 2-76: IPV6 MULTICAST DESTINATION MASK

Destination Ports	Record Bit Field
Ports 10 through 0	MAC [42-32]

Example: All IPv6 multicast frames in VLAN 12 with MAC 333300112233 are to be forwarded to ports 3, 8, and 9.

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

VID = 12
MAC = 0x030800112233
ENTRY_TYPE = 3
VALID

1
DEST_IDX = 0

Port, VLAN, and Domain Filter

The following table lists the registers associated with the ageing filter.

TABLE 2-77: VID/PORT/DOMAIN FILTERS

Register	Description	Replication
ANAGEFIL	Port, VLAN, and domain filter for limiting the target for aging and search operations on MAC table.	None

The ANAGEFIL register can be used to only hit specific VLANs or ports 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 2.9.1.3, CPU Commands.

When two or more filters are enabled at the same time, for example, port and domain, all conditions must be fulfilled before the operation (aging, GET_NEXT) is carried out.

2.9.1.7 Shared VLAN Learning

The following table lists the location of the Filter Identifier (FID) used for shared VLAN learning.

TABLE 2-78: FID DEFINITION REGISTERS

Register	Description	Replication
IS1_ACTION.FID_SEL	Specifies the use of IS1_ACTION.FID_VAL for the DMAC lookup, the SMAC lookup, or for both lookups.	Per IS1 entry
IS1_ACTION.FID_VAL	FID value used when FID_SEL>0. This FID takes precedence.	Per IS1 entry
ANA::FID_MAP.FID_C_VAL	VID-to-FID mapping table. 64 FIDs are programmable. FID_C_VAL=0 effectively disables the mapping implying that FID_C_VAL=VID.	Per VID
AGENCTRL.FID_MASK	Combines multiple VIDs in the MAC table.	None

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. Bit 13 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.

2.9.1.8 Learn Limit

The following table lists the registers associated with controlling the number of MAC table entries per port.

TABLE 2-79: LEARN LIMIT DEFINITION REGISTERS

Register	Description	Replication
ENTRYLIM	Configures maximum number of unlocked entries in the MAC table per ingress port.	Per port
PORT_CFG.LIMIT_CPU	If set, learn frames exceeding the limit are copied to the CPU.	Per port
PORT_CFG.LIMIT_DROP	If set, learn frames exceeding the limit are discarded.	Per port

TABLE 2-79: LEARN LIMIT DEFINITION REGISTERS (CONTINUED)

Register	Description	Replication
LEARNDISC	The number of MAC table entries that could not be learned due to a lack of storage space.	None

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.

2.9.2 VLAN TABLE

The following table lists the registers associated with the VLAN Table.

TABLE 2-80: VLAN TABLE ACCESS

Register	Description	Replication
VLANTIDX	VID to access, and VLAN flags.	None
VLANACCESS	VLAN port mask for VID and command for access.	None

The analyzer has a VLAN table that contains information about the members of each of the 4096 VLANs. The following table lists fields for each entry in the VLAN table.

TABLE 2-81: FIELDS IN THE VLAN TABLE

Field	Bits	Description
VLAN_PORT_MASK	11	One bit for each port. Set if port is member of VLAN. The CPU port is always a member of all VLANs.
VLAN_MIRROR	1	Mirror frames received in the VLAN. See Section 2.9.3.8, 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.

By default, all ports are members of all VLANs. This default can be changed through a CPU command. The following table 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 2-82: 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.

TABLE 2-82: VLAN TABLE COMMANDS (CONTINUED)

Command	Purpose	Use
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
IDLE	Indicate that VLAN table is ready for new command.	No preload required.

2.9.3 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 te 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.

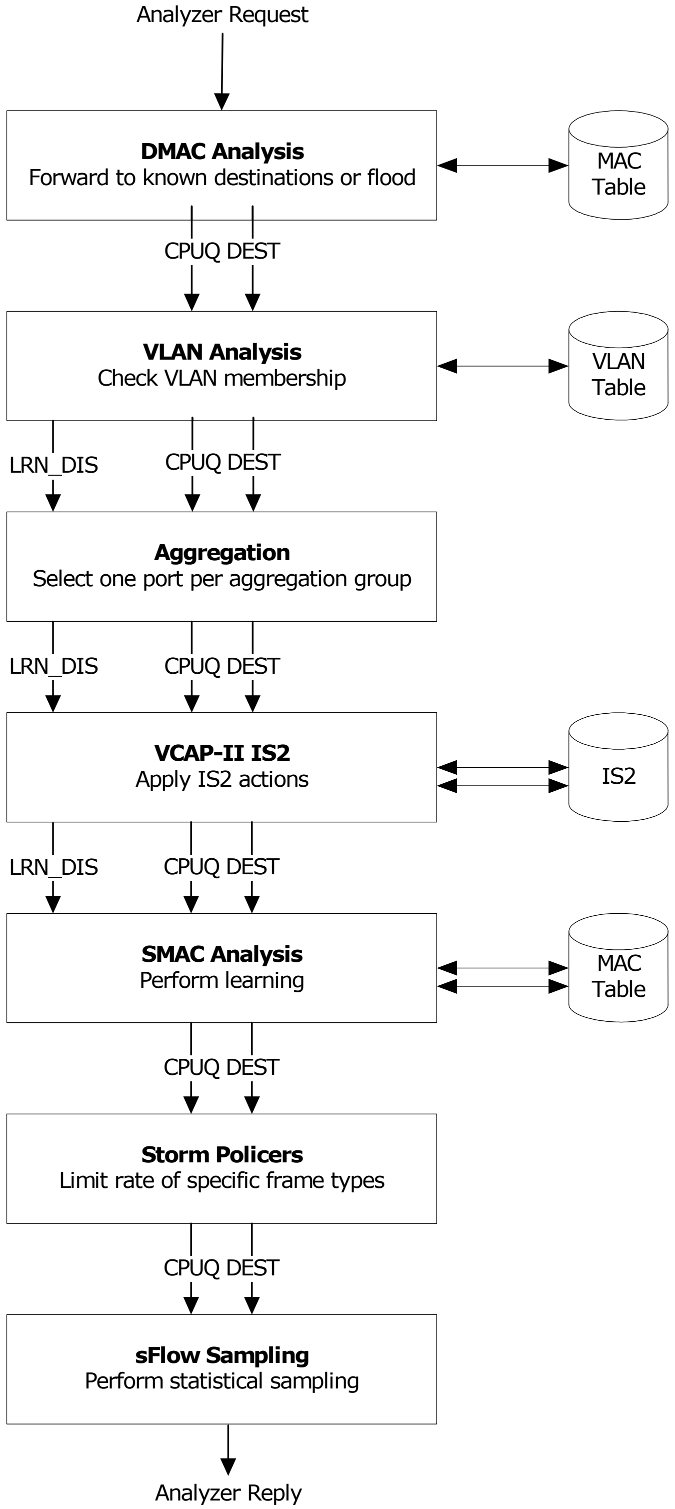
In addition to the forwarding decision, the analyzer determines which frames are subject to learning (also known as learn frames). Learn frames trigger insertion of a new entry in the MAC table or update of an existing entry. Learning is presented as part of the forwarding, because in some cases, learning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

The following illustration shows the configuration steps in the analyzer.

FIGURE 2-28: ANALYSIS STEPS



2.9.3.1 DMAC Analysis

During the DMAC analysis step, the (DMAC, VID) pair is looked up in the MAC table to get the first input to the calculation of the destination port set. For more information about the MAC table, see Section 2.9.1, MAC Table.

The following table lists the registers associated with the DMAC analysis step.

TABLE 2-83: DMAC ANALYSIS REGISTERS

Register	Description	Replication
FLOODING.FLD_UNICAST	Index into the PGID table used for flooding of unicast frames.	None
FLOODING.FLD_BROADCAST	Index into the PGID table used for flooding of broadcast frames.	None
FLOODING.FLD_MULTICAST	Index into the PGID table used for flooding of multicast frames, not flooded by the IPMC flood masks.	None
FLOODING_IPMC.FLD_MC4_CTRL	Index into the PGID table used for flooding of IPv4 multicast control frames.	None
FLOODING_IPMC.FLD_MC4_DATA	Index into the PGID table used for flooding of IPv4 multicast data frames.	None
FLOODING_IPMC.FLD_MC6_CTRL	Index into the PGID table used for flooding of IPv6 multicast control frames.	None
FLOODING_IPMC.FLD_MC6_DATA	Index into the PGID table used for flooding of IPv6 multicast data frames.	None
PGID[63:0]	Destination and flooding masks table.	64
AGENCTRL. IGNORE_DMAC_-FLAGS	Controls the use of MAC table flags from (DMAC, VID) entry and flooding flags.	None
CPUQ_CFG	Configuration of CPU extraction queues	None

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.

If a match is found, the entry is returned and DEST is determined based on the MAC table entry. For more information, see Section 2.9.1, MAC Table.

If an entry is found in the MAC table entry of ENTRY_TYPE `0` or `1` and the CPU port is set in the PGID pointed to by the MAC table entry, CPU extraction queue PGID.DST_PGID is added to the CPUQ.

If an entry is not found for the (DMAC, VID) in the MAC table, the frame is flooded. The forwarding decision is set to one of the seven flooding masks defined in ANA::FLOODING or ANA::FLOODING_IPMC, based on one of the flood type definitions listed in the following table.

TABLE 2-84: FORWARDING DECISIONS BASED ON FLOOD TYPE

Frame Type	Condition

IPv4 multicast data	DMAC = 0x01005E000000 to 0x01005E7FFFFF
	EtherType = IPv4
	IP protocol is not IGMP
	IPv4 DIP outside 224.0.0.x

IPv6 multicast data	DMAC = 0x333300000000 to 0x3333FFFFFFFF
	EtherType = IPv6
	IPv6 DIP outside 0xFF02::/16

IPv4 multicast control	DMAC = 0x01005E000000 to 0x01005E7FFFFF
	EtherType = IPv4
	IP protocol is not IGMP
	IPv4 DIP inside 224.0.0.x

IPv6 multicast control	DMAC = 0x333300000000 to 0x3333FFFFFFFF
	EtherType = IPv6
	IPv6 DIP inside 0xFF02::/16

TABLE 2-84: FORWARDING DECISIONS BASED ON FLOOD TYPE (CONTINUED)

Frame Type	Condition
Broadcast	DMAC = 0xFFFFFFFFFFFF non-IPv4-multicast-data non-IPv6-multicast-data non-IPv4-multicast-control non-IPv6-multicast-control
Multicast	Bit 40 in DMAC = 1 non-broadcast non-IPv4-multicast-data non-IPv6-multicast-data non-IPv4-multicast-control non-IPv6-multicast-control
Unicast	Bit 40 in DMAC = 0

In addition, the MAC table 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.

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 2-31.

2.9.3.2 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 2.9.2, VLAN Table.

The following table lists the registers associated with VLAN analysis.

TABLE 2-85: VLAN ANALYSIS REGISTERS

Register	Description	Replication
VLANMASK	If PPORT is set in this mask, and PPORT is not member of the VLAN to which the frame is classified, DEST is cleared. This is also called VLAN ingress filtering.	None
PORT_CFG.RECV_ENA	If this bit is cleared for PPORT, forwarding from this port to other front ports is disabled, and DEST is cleared.	Per port
PGID[91:80]	Source port mask. Port mask per port, which specifies allowed destination ports for frames received on PPORT. By default, a port can forward to all other ports except itself.	Per port
ISOLATED_PORTS	Private VLAN mask. Isolated ports are cleared in this mask.	None
COMMUNITY_PORTS	Private VLAN mask. Community ports are cleared in this mask.	None
ADVLEARN.VLAN_CHK	If set and VLAN ingress filtering clears DEST, then SMAC learning is disabled.	None

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 MAC table has VLAN_IGNORE set, or if there is no match in 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.

Finally, 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 ADVLEARN.VLAN_CHK is set.

2.9.3.3 Aggregation

During the aggregation step, link aggregation is handled. The following table lists the registers associated with aggregation.

TABLE 2-86: ANALYZER AGGREGATION REGISTERS

Register	Description	Replication
PGID[79:64]	Aggregation mask table.	16

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.

For more information about link aggregation, see Section 4.3.8, Link Aggregation.

2.9.3.4 VCAP Action Handling

VCAP IS2 actions are processed during the VCAP IS2 action handling step. The following table lists the processing of the VCAP actions. The order of processing is from top to bottom.

TABLE 2-87: 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 2.10, 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) are disabled. See Section 2.10, Policers.
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 2.9.3.8, Mirroring.

2.9.3.5 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. Either the B-MAC table or the C-MAC table is searched. The learning part is skipped if the LRN_DIS flag was set by any of the previous steps.

The following table lists the registers associated with SMAC learning.

TABLE 2-88: SMAC LEARNING REGISTERS

Register	Description	Replication
PORT_CFG.LEARN_ENA	If cleared for PPORT, learning is skipped (that is, LEARNAUTO, LEARNCPU, LEARNDROP, LIMIT_CPU, LIMIT_DROP, LOCKED_PORTMOVE_CPU, and LOCKED_PORTMOVE_DROP are ignored).	Per port
PORT_CFG.LEARNAUTO	If set for PPORT, hardware-based learning is performed.	Per port
PORT_CFG.LEARNCPU	If set for PPORT, learn frames are copied to the CPU.	Per port
PORT_CFG.LEARNDROP	If set for PPORT, the CPU drops or forwards learn frames.	Per port
PORT_CFG.LIMIT_CPU	If set for PPORT, learn frames for which PPORT exceeds the port's limit are copied to the CPU.	Per port
PORT_CFG.LIMIT_DROP	If set for PPORT, learn frames for which PPORT exceeds the port's limit are discarded.	Per port
PORT_CFG.LOCKED_PORTMOVE_CPU	If set for PPORT, frames triggering a port move of a locked entry are copied to the CPU.	Per port
PORT_CFG.LOCKED_PORTMOVE_DROP	If set for PPORT, frames triggering a port move of a locked entry are discarded.	Per port

TABLE 2-88: SMAC LEARNING REGISTERS (CONTINUED)

Register	Description	Replication
AGENCTRL.IGNORE_SMAC_FLAGS	Controls the use of the MAC table flags from (SMAC, VID) entry.	None

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 2.10, Policers.

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

2.9.3.5.1 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.

2.9.3.5.2 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.

2.9.3.5.3 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.

2.9.3.6 Storm Policers

The storm policers are activated during the storm policers step. The following table lists the registers associated with storm policers.

TABLE 2-89: STORM POLICER REGISTERS

Register	Description	Replication
STORMLIMIT_CFG	Enables policing of various frame types.	4
STORMLIMIT_BURST	Configures maximum allowed rates of the different frame types.	None

The analyzer contains four storm policers that can limit the maximum allowed forwarding frame rate for various frame types. 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 several policers, and the frame is discarded if any of the activated policers exceed a configured rate.

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

The following table lists the available policers.

TABLE 2-90: STORM POLICERS

Policer	Description
Broadcast	Flooded frames with DMAC = 0xFFFFFFFFFFFF.
Multicast	Flooded frames with DMAC bit 40 set, except broadcasts.
Unicast	Flooded frames with DMAC bit 40 cleared.
Learn	Learn frames copied or redirected to the CPU due to learning (LOCKED_PORTMOVE_CPU, LIMIT_CPU, LEARNCPU).

For each of the 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 policer affects the forwarding decision. The options are:
- When policing, clear CPUQ.
- When policing, clear DEST.
- When policing, clear DEST and CPUQ.

Frames where the DMAC lookup returned a PGID with the CPU port set are always forwarded to the CPU even when the frame is policed by the storm policers. For more information, see Section 2.9.3.1, DMAC Analysis.

2.9.3.7 sFlow Sampling

This process step handles sFlow sampling. The following table lists the registers associated with sFlow sampling.

TABLE 2-91: SFLOW SAMPLING REGISTERS

Register	Description	Replication
SFLOW_CFG	Configures sFlow samplers (type and rates).	Per port
CPUQ_CFG.CPUQ_SFLOW	CPU extraction queue for sFLow sampled frames.	None

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.

2.9.3.8 Mirroring

This processing step handles mirroring. The following table lists the registers associated with mirroring.

TABLE 2-92: MIRRORING REGISTERS

Register	Description	Replication
ADVLEARN.LEARN_MIRROR	For learn frames, ports in this mask (mirror ports) are added to DEST.	None
AGENCTRL.MIRROR_CPU	Mirror all frames forwarded to the CPU port module.	None
PORT_CFG.SRC_MIRROR_ENA	Mirror all frames received on an ingress port (ingress port mirroring).	Per port
EMIRRORMASK	Mirror frames that are to be transmitted on any ports set in this mask (egress port mirroring).	None
VLANTIDX.VLAN_MIRROR	Mirror all frames classified to a specific VID.	Per VLAN
IS2_ACTION.MIRROR_ENA	Mirror when an IS2 action is hit.	Per VCAP IS2 entry
MIRRORPORTS	When mirroring a frame, ports in this mask are added to DEST.	None
AGENCTRL.CPU_CPU_KILL_ENA	Clear the CPU port if source port is the CPU port and the CPU port is set in DEST.	None

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 2.9.3.5, SMAC Analysis.

For more information about mirroring, see Section 4.3.10, Mirroring.

Finally, if AGENCTRL.CPU_CPU_KILL_ENA is set, the CPU port is removed if the ingress port is the CPU port itself. This is similar to source port filtering done for front ports and prevents the CPU from sending frames back to itself.

2.9.4 ANALYZER MONITORING

Miscellaneous events in the analyzer can be monitored, which can provide an understanding of the events during the processing steps. The following table lists the registers associated with analyzer monitoring.

TABLE 2-93: ANALYZER MONITORING

Register	Description	Replication
ANMOVED	ANMOVED[n] is set when a known station has moved to port n.	None
ANEVENTS	Sticky bit register for various events.	None
LEARNDISC	The number of learn events that failed due to a lack of storage space in the MAC table.	None

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.

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.

2.10 Policers

The device has 192 policers that can be allocated to ingress ports, QoS classes per port, and VCAP IS2 entries. 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 policer and one QoS 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 2.9.3.6, Storm Policers and Section 2.12, Scheduler and Shapers.

The following table lists the registers associated with policer control.

TABLE 2-94: POLICER CONTROL REGISTERS

Register	Description	Replication
ANA:PORT:POL_CFG	Enables use of port and QoS policers	Per port
ANA:POL:POL_PIR_CFG	Configures the policer’s peak information rate	384
ANA:POL:POL_CIR_CFG	Configures the policer’s committed information rate	384
ANA:POL:POL_MODE_CFG	Configures the policer’s mode of operation	384
ANA:POL:POL_PIR_STATE	Current state of the peak information rate bucket	384
ANA:POL:POL_CIR_STATE	Current state of the committed information rate bucket	384
ANA:PORT:POL_FLOWC	Flow control settings	Per port
ANA::POL_HYST	Hysteresis settings	None

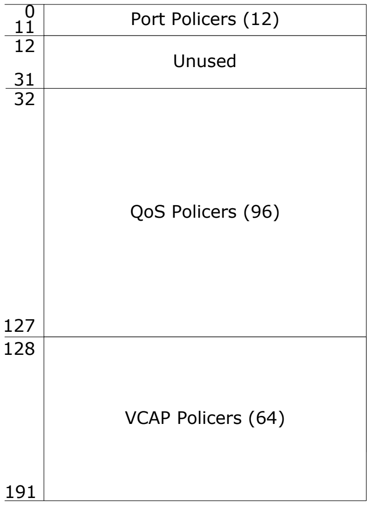
2.10.1 POLICER ALLOCATION

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 32 + 8x ‘p’ + ‘q’. Each of the eight per-port QoS classes per port can be assigned to its own policer.
- VCAP IS2 policers. Policers 128 through 191 can be allocated to VCAP policing. The action IS2_ACTION.POLICE_IDX points to the policer that is used.

The policer pool layout is illustrated in the following drawing.

FIGURE 2-29: POLICER POOL LAYOUT



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.

Finally, 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. The VCAP can point to any unused policer in addition to the dedicated VCAP policers.

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 the port policing and QoS class policing. This is done with the action POLICE_V-CAP_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.

2.10.2 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. If higher bandwidths are required, the policer for 2.5G ports must be disabled.

- Committed Burst Size (CBS). Specified in POL_CIR_CFG.CIR_BURST in steps of 4 kilobytes. Maximum is 252 kilobytes.
- Excess Information Rate (EIR). Specified in POL_PIR_CFG.PIR_RATE in steps of 33.3 kbps. Maximum rate is 1.09 Gbps. If higher bandwidths are required, the policer for 2.5G ports must be disabled.
- Excess Burst Size (EBS). Specified in POL_PIR_CFG.PIR_BURST in steps of 4 kilobytes. Maximum is 252 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.

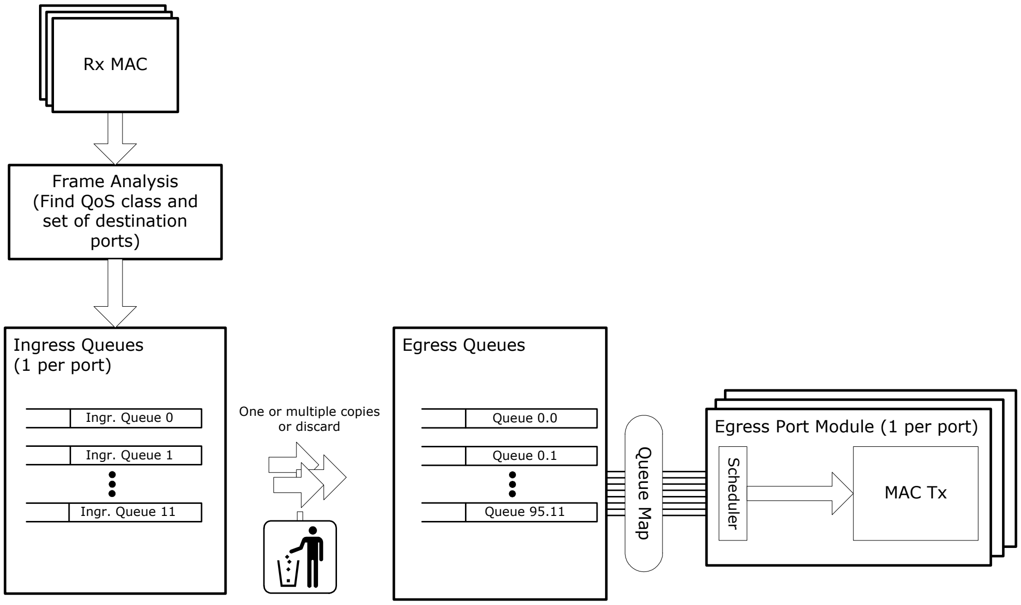
2.11 Shared Queue System

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 224 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. The transfer from ingress to egress is extremely efficient with a transfer time of 12.8 ns per frame copy (equivalent to a transfer rate from ingress to egress of about 50 Gbps for 64-byte frames and about 1 terabit per second (Tbps) for 1518-byte frames). Each egress port module has a scheduler that selects between the egress queues when transmitting frames.

The following illustration shows the shared queue system.

FIGURE 2-30: QUEUE SYSTEM OVERVIEW


Rx MAC
Frame Analysis (Find QoS class and set of destination ports)
Ingress Queues (1 per port)
Ingr. Queue 0
Ingr. Queue 1
Ingr. Queue 11
One or multiple copies or discard
Egress Queues
Queue 0.0
Queue 0.1
Queue 95.11
Queue Map
Egress Port Module (1 per port)
Scheduler
MAC Tx

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 2.11.7, Ingress Pause Request Generation.

2.11.1 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 2.11.7, 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. The following table lists the reservation watermarks.

TABLE 2-95: RESERVATION WATERMARKS

Register	Description	Replication
BUF_Q_RSRV_E	Configures the reserved amount of egress buffer per QoS class per egress port.	Per QoS class per egress port
BUF_P_RSRV_E	Configures the reserved amount of egress buffer shared among the egress port's allocated egress queues.	Per egress port
BUF_Q_RSRV_I	Configures the reserved amount of egress buffer per ingress port per QoS class across all egress ports.	Per ingress port per QoS class
BUF_P_RSRV_I	Configures the reserved amount of egress buffer per ingress port shared among the eight QoS classes.	Per ingress port

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 following table lists the sharing watermarks.

TABLE 2-96: SHARING WATERMARKS

Register	Description	Replication
BUF_PRIO_SHR_E	Configures how much of the shared memory area that egress frames with the given QoS class are allowed to use.	Per QoS class
BUF_COL_SHR_E	Configures how much of the shared memory area that egress frames with the given drop precedence level are allowed to use.	Per drop precedence level
BUF_PRIO_SHR_I	Configures how much of the shared memory area that ingress frames with the given QoS class are allowed to use.	Per QoS class
BUF_COL_SHR_I	Configures how much of the shared memory area that ingress frames with the given drop precedence level are allowed to use.	Per drop precedence level

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.

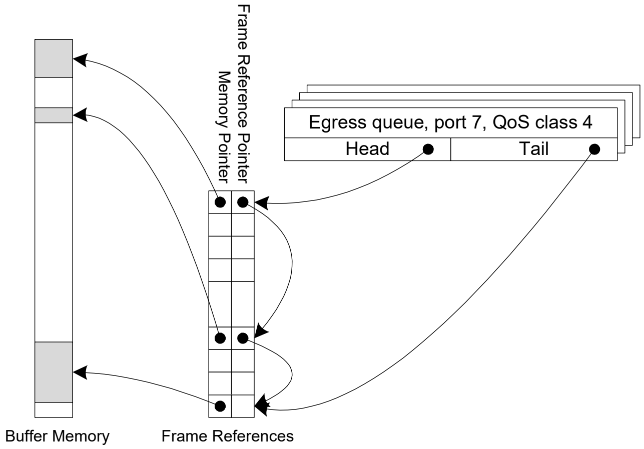
- Setting BUF_Q_RSRV_E(egress port = 7, QoS class = 4) to 2 kilobytes guarantees that traffic destined for port 7 classified to QoS class 4 have room for 2 kilobytes of frame data before frames can get discarded.
- Setting BUF_Q_RSRV_I(ingress port = 7, QoS class = 4) to 2 kilobytes guarantees that traffic received on port 7 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 7) to 10 kilobytes guarantees that traffic received on port 7 have room for 10 kilobytes of data before frames can get discarded.
- The three reservations above reserve 14 kilobytes of memory in total (2 + 2+ 10 kilobytes) for port 7. If the same reservations are made for all ports, there are 224 – 11 × 14 = 70 kilobytes left for sharing. If the sharing watermarks are all set to 70 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 70 kilobytes and the other watermarks BUF_PRIO_SHR_E(QoS class = 0:6) to 20 kilobytes guarantees that traffic classified to QoS class 7 has 50 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.

2.11.2 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. The following illustrations shows how the frame references are used for creating the queue structure.

FIGURE 2-31: FRAME REFERENCE



The shared queue system holds a table of 1911 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. For more information, see Table 2-95 and Table 2-96.

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.

2.11.3 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.

2.11.4 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 2-97: 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.

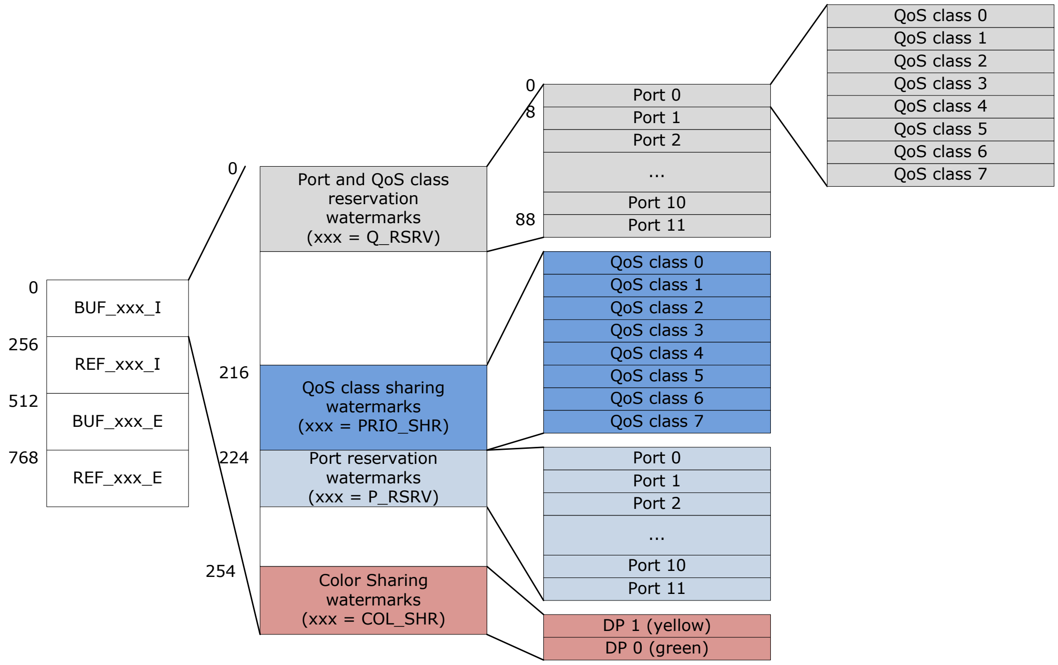
TABLE 2-97: WATERMARK CONFIGURATION EXAMPLE (CONTINUED)

Watermark	Value	Comment
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 110 kilobytes of the total 224 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	42 kilobytes to 63 kilobytes	The different QoS classes are cut-off with 3 kilobytes distance (42, 45, 48, 51, 54, 57, 60, and 63 kilobytes). This gives frames with higher QoS classes a larger part of the shared buffer area. Effectively, this means that the burst capacity is 52 kilobytes for frames belonging to QoS class 0 and up to 73 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.
REF_PRIO_SHR_E REF_PRIO_SHR_I	1350 - 1700	The different QoS classes are cut-off with a distance of 50 frame references (1350, 1400, 1450, 1500, 1550, 1600, 1650, and 1700). This gives frames with higher QoS classes a larger part of the shared reference area.

2.11.5 WATERMARK PROGRAMMING AND CONSUMPTION MONITORING

The watermarks previously described are all found in the SYS::RES_CFG register. The register is replicated 1024 times. The following illustration the organization.

FIGURE 2-32: 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.

2.11.6 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 the following table.

TABLE 2-98: 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.

2.11.7 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 2.2.1, 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.

2.11.8 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

2.11.9 TEST UTILITIES

This section describes some of test utilities that 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.

2.11.10 ENERGY EFFICIENT ETHERNET

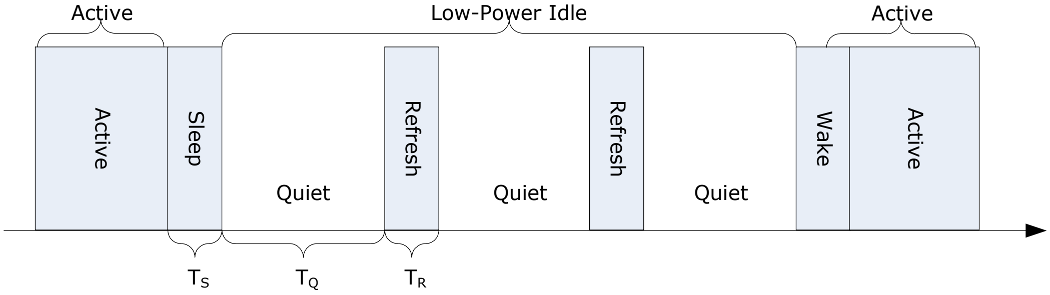
This section provides information about the functions of Energy Efficient Ethernet in the shared queue system. The following table lists the registers associated with Energy Efficient Ethernet.

TABLE 2-99: ENERGY EFFICIENT ETHERNET CONTROL REGISTERS

Register	Description	Replication
DEV::EEE_CFG	Enables configuration of Energy Efficient Ethernet. Status bit indicating that egress port is in the LPI state.	Per port
QSYS::EEE_CFG	Configures fast queues.	Per port
QSYS::EEE_THRES	Configures bytes and frame thresholds.	None

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 2-33: 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).

2.12 Scheduler and Shapers

The following table lists the registers associated with the scheduler and egress shaper control.

TABLE 2-100: SCHEDULER AND EGRESS SHAPER CONTROL REGISTERS

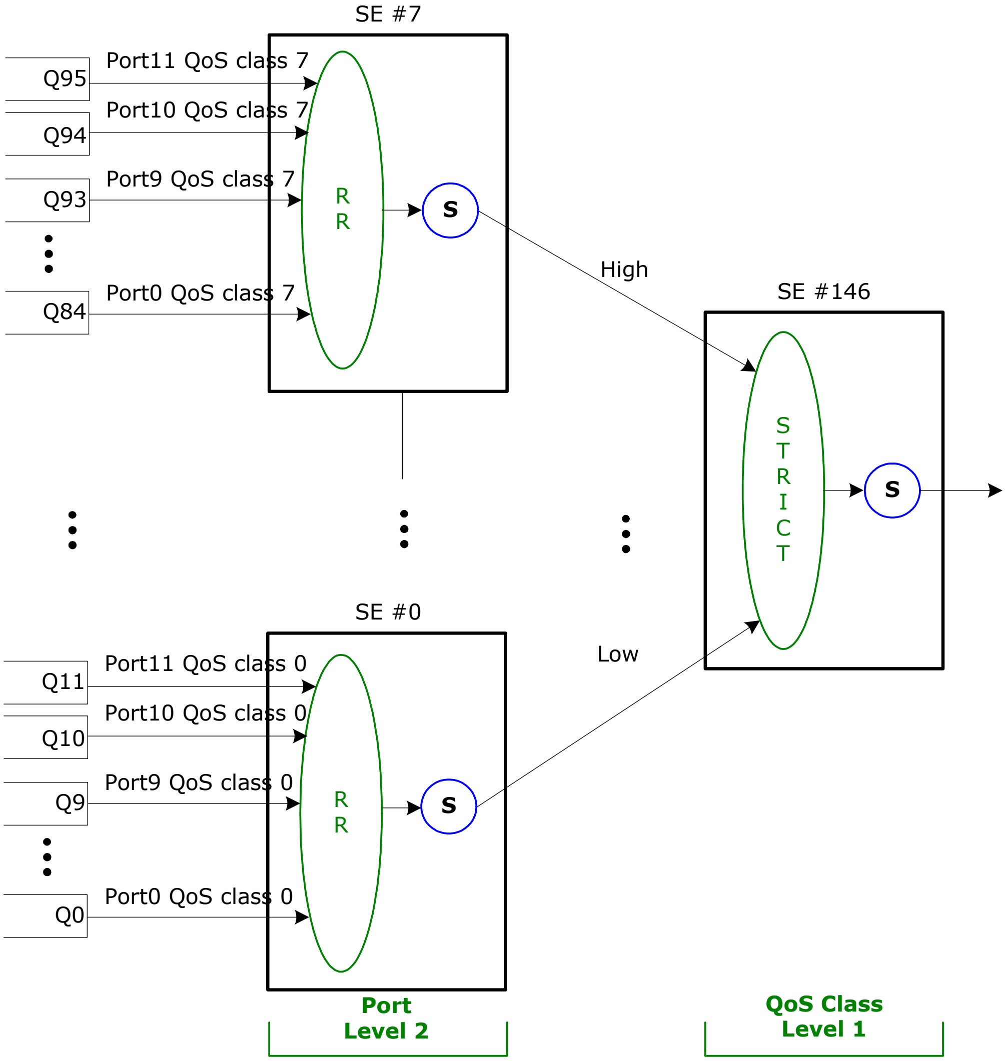
Register	Description	Replication
QSYS::SHAPER_CFG	Configuration of egress shaper's rate and burst.	158
QSYS::SE_CFG	Configuration of the scheduling algorithm.	158
QSYS::SE_DWRR_CFG	Configuration of DWRR scheduler's costs.	158
QSYS::SHAPER_STATE	Status of the shaper bucket.	158

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 12 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.

The following illustration provides an overview of the egress scheduling system for egress port 0.

FIGURE 2-34: EGRESS SCHEDULER PORT 0



Each egress port features a similar egress scheduler. The following table lists which scheduler elements are used by the different ports.

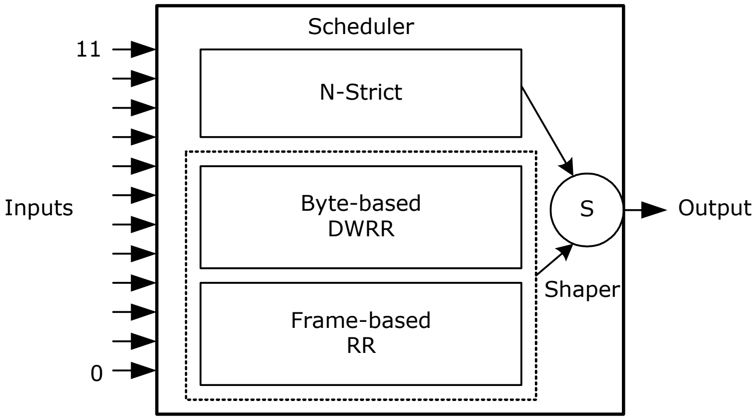
TABLE 2-101: SCHEDULER ELEMENTS NUMBERING

Egress Port	Scheduler Elements - Level 1	Scheduler Elements - Level 2
0	146	0 through 7
1	147	8 through 15
2	148	16 through 23
3	149	24 through 31
4	150	32 through 39
5	151	40 through 47
6	152	48 through 55
7	153	56 through 63
8	154	64 through 71
9	155	72 through 79
10	156	80 through 87
11 (CPU)	157	88 through 95

2.12.1 SCHEDULER ELEMENT

The following illustration shows the scheduler element.

FIGURE 2-35: SCHEDULER ELEMENT



By default, the scheduler operates in strict priority between all 12 inputs. The inputs are searched in the following prioritized order: Input 11 has highest priority followed by 10, 9, 8, 7, 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.

2.12.2 EGRESS SHAPERS

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

- Maximum rate – Specified in SHAPER_CFG.RATE in steps of 100160 bps. Maximum is 3.282 Gbps.
- Maximum burst size – Specified in SHAPER_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.

2.12.3 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%

2.12.4 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.

2.12.5 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.

2.12.5.1 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.

The following table lists the DWRR configuration and the resulting egress bandwidth for the various queues.

TABLE 2-102: 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**

2.12.5.2 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.

The following table lists the resulting egress bandwidth for the various queues.

TABLE 2-103: EXAMPLE OF STRICT AND WORK-CONSERVING SHAPING

Queue	Result: Egress Bandwidth
Q0	50 Mbps
Q1	50 Mbps

TABLE 2-103: EXAMPLE OF STRICT AND WORK-CONSERVING SHAPING (CONTINUED)

Queue	Result: Egress Bandwidth
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**

2.13 Rewriter

The device includes a rewriter common for all ports that determines how the egress frame is edited before transmission. The rewriter performs the following editing:

- VLAN editing; tagging of frames and remapping of PCP and DEI.
- DSCP remarking; rewriting the DSCP value in IPv4 and IPv6 frames based on classified DSCP value.
- FCS updating.
- Precision Time Protocol time stamp updating.

Each port module including the CPU port module (CPU port 11 and CPU port 12) has its own set of configuration in the rewriter. Each frame is handled by the rewriter one time per destination port.

Most rewriting functions in the rewriter are independent of each other and can co-exist. For instance, a frame can be VLAN tagged while at the same time being DSCP remarked. However, precision time protocol time stamp updating and DSCP remarking are mutually exclusive with the former taking precedence. If an IS2 action returns any PTP actions then DSCP remarking is automatically disabled for the frame.

2.13.1 VLAN EDITING

The following table lists the registers associated with VLAN editing.

TABLE 2-104: VLAN EDITING REGISTERS

Register	Description	Replication
PORT_VLAN_CFG	Port VLAN for egress port. Used for untagged set	Per port
TAG_CFG	Tagging rules for port tag	Per port
PORT_CFG.ESO_ENA	Enable lookups in ES0	Per port
PCP_DEI_QOS_MAP_CFG	Mapping table Maps DP level and QoS class to new PCP and DEI values	Per port per QoS per DP

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.

2.13.1.1 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.

2.13.1.2 ES0 Lookup

For each of the frame's destination ports, VCAP ES0 is looked up using the ES0 key. See Section 2.8.4, VCAP ES0 for more information about ES0. The action from an ES0 hit is used in the following to determine the frame's VLAN editing.

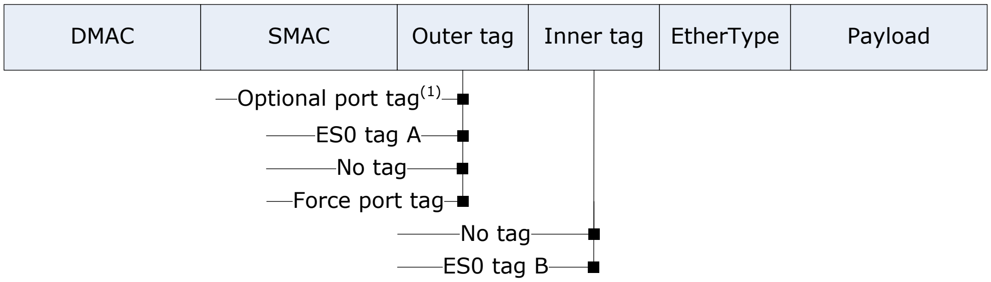
2.13.1.3 VLAN Push Decision

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.

The following illustration shows an overview of the available tagging options.

FIGURE 2-36: TAGGING OVERVIEW



(1) Port tag is controlled by REW:PORT:TAG_CFG.TAG_CFG.

The following table 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 2-105: 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.

TABLE 2-105: TAGGING COMBINATIONS (CONTINUED)

ES0_ACTION	TAG_CFG	Tagging action

PUSH_OUTER_TAG=3
PUSH_INNER_TAG=0
	Don't care	No tags are pushed. This overrules port settings in TAG_CFG.

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.

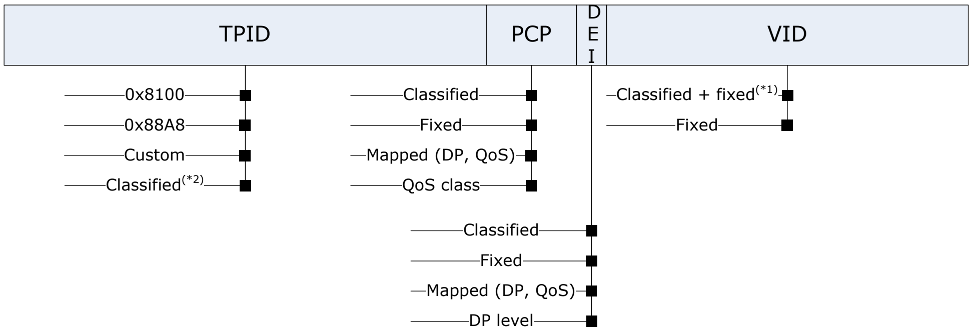
PUSH_OUTER_TAG=2
PUSH_INNER_TAG=1
	Don't care	Port tag is pushed as outer tag. This overrules port settings in TAG_CFG.
		ES0 tag B is pushed as inner tag.

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.

2.13.1.4 Constructing VLAN Tags

When pushing a VLAN tag, the contents of the tag header, including the TPID, are 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 reference>.

FIGURE 2-37: TAG CONSTRUCTION (PORT TAG, ES0 TAG A, ES0 TAG B)



(*1) For port tag, this is classified only.
(*2) Use 0x8100 if classified tag type is 0x8100, otherwise custom.

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.

2.13.1.5 VLAN Pushing

In the final VLAN editing step, the VLAN tags derived from the previous steps are pushed to the frame.

2.13.2 DSCP REMARKING

The following table lists the registers associated with DSCP remarking.

TABLE 2-106: DSCP REMARKING REGISTERS

Register	Description	Replication
DSCP_CFG	Selects how the DSCP remarking is done	Per port
DSCP_REMAP_CFG	Mapping table from DSCP to DSCP for DP level = 0.	None
DSCP_REMAP_DP1_CFG	Mapping table from DSCP to DSCP for DP level = 1.	None

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.

DSCP remarking is not possible for frames where PTP time stamps are also generated and is automatically disabled.

2.13.3 FCS UPDATING

The following table lists the registers associated with FCS updating.

TABLE 2-107: FCS UPDATING REGISTERS

Register	Description	Replication
PORT_CFG.FCS_UPDATE_NONCPU_CFG	FCS update configuration for non-CPU injected frames.	Per port
PORT_CFG.FCS_UPDATE_CPU_ENA	FCS update configuration for CPU injected frames.	Per port

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 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.

2.13.4 PTP TIME STAMPING

The following table lists the registers associated with PTP time stamping.

TABLE 2-108: PTP TIME STAMPING REGISTERS

Register	Description	Replication
REW:PORT:PTP_CFG	PTP Configuration of egress port	Per port
REW:PORT:PTP_DLY1_CFG	Egress delay configuration	Per port

The rewriter can do various different PTP time stamping in the egress frame:

Residence time. Adds the frame’s residence time through the switch to the PTP correction field (byte-offset 8).

Add/subtract. Adds the frame’s Tx time to the correction field (byte-offset 8) or subtract the frame’s Rx time from the correction field.

Time-of-day. Samples and writes the 80-bit time-of-day into the PTP origin time stamp (byte-offset 34).

Set reserved bytes. Writes the RxTime into the PTP reserved bytes (byte-offset 16).

Clear reserved bytes. Writes zero into the PTP reserved bytes (byte-offset 16).

This can be done for PTP frames over IEEE802.3/Ethernet, PTP frames over UDP over IPv4, and PTP frames over UDP over IPv6. In addition, frames can be VLAN tagged. The rewriter automatically finds the correct location of the PTP header in the frame.

The rewriter takes the following frame properties from the analyzer as input.

- IS2 rewriter action (IS2_ACTION.REW_OP[2:0]), which can be either one-step, two-step, or origin.
- The frame's Rx time stamp, adjusted for I/O path delays and ingress delays according to ingress actions.
- IS2 rewriter actions for one-step PTP: Add/subtract mode, egress delay adjustment.
- Ingress backplane mode ANA:PORT:PTP_CFG.PTP_BACKPLANE_MODE from the frame's ingress port.

In addition, the following egress port properties are used.

- Egress port delay (REW:PORT:PTP_DLY1_CFG).
- Egress backplane mode (REW:PORT:PTP_CFG.PTP_BACKPLANE_MODE).

The following table shows the resulting rewriter actions for the one-step PTP action, depending on the add/subtract mode and ingress and egress backplane mode settings.

TABLE 2-109: PTP TIME STAMPING FOR ONE-STEP PTP

PTP Action (from IS2)	Add/Subtract	Ingress Backplane	Egress Backplane	PTP time stamp Rewriter Actions
One-step	Disabled	Disabled	Disabled	Residence time: Adds frame's residence time to PTP correction field. Egress port delay is added to correction field if specified by IS2.
One-step	Disabled	Disabled	Enabled	Sets reserved bytes: Writes frame's Rx time stamp into the reserved bytes.
One-step	Disabled	Enabled	Disabled	Residence time: Adds frame's residence time to PTP correction field. Egress port delay is added to correction field if specified by IS2. Clear reserved bytes.
One-step	Disabled	Enabled	Enabled	Does not apply.
One-step	Enabled	Disabled	Disabled	Residence time: Adds frame's residence time to PTP correction field. Egress port delay is added to correction field if specified by IS2.
One-step	Enabled	Disabled	Enabled	Add/subtract: Subtracts frame's Rx time stamp from PTP correction field.
One-step	Enabled	Enabled	Disabled	Add/subtract: Adds frame's Tx time stamp to PTP correction field. Egress port delay is added to correction field if specified by IS2. Clear reserved bytes.
One-step	Enabled	Enabled	Enabled	Does not apply.

All actions associated with one-step or origin can be disabled per egress port through REW:PORT:PTP_CFG.PTP_1-STEP_DIS.

The following table shows the resulting rewriter actions for the two-step PTP action, depending on the ingress and egress backplane mode settings.

TABLE 2-110: PTP TIME STAMPING FOR TWO-STEP PTP

PTP action (from IS2)	Ingress Backplane	Egress Backplane	PTP time stamp Rewriter Actions
Two-step	Disabled	Disabled	Saves Tx time stamp in PTP time stamp queue.
Two-step	Disabled	Enabled	Sets reserved bytes: Write the frame’s Rx time stamp into the reserved bytes.
Two-step	Enabled	Disabled	Saves Tx time stamp in PTP time stamp queue. Clear reserved bytes.
Two-step	Enabled	Enabled	Does not apply.

All actions associated with two-step can be disabled per egress port through REW:PORT:PTP_CFG.PTP_2STEP_DIS.

The following table shows the resulting rewriter actions for the origin PTP action, depending on the ingress and egress backplane mode settings.

TABLE 2-111: PTP TIME STAMPING FOR ORIGIN PTP

PTP Action (from IS2)	Ingress Backplane	Egress Backplane	PTP time stamp Rewriter Actions
Origin	Disabled	Disabled	Time-of-day: Write the time-of-day into the PTP origin time stamp field.
Origin	Disabled	Enabled	None.
Origin	Enabled	Disabled	Time-of-day: Write the time-of-day into the PTP origin time stamp field.
Origin	Enabled	Enabled	Does not apply.

The following describes each of the PTP time stamping rewriter actions.

Residence Time. The frame’s residence time is calculated as Tx time stamp - Rx time stamp. For information about how the Rx and Tx time stamps are derived, see Section 2.2.1.8, Rx and Tx Time Stamps. The residence time is adjusted for an optional egress delay (REW:PORT:PTP_DLY1_CFG) enabled through IS2_ACTION.REW_OP[5] = 1. The resulting residence time is added to the correction field value in the PTP header. The result is written back into the frame.

Add/subtract. Add the frame’s Tx time to the correction field or subtract the frame’s Rx time from the correction field. This is used in backplane applications. This mode requires a rollover protection mode to handle the situation where the nanosecond counter rolls over during the backplane transfer. The rollover protection mode is configured in SYS::PTP_CFG.PTP_CF_ROLL_MODE and must be the same in both the ingress and the egress backplane unit.

Time-of-Day. The time-of-day consists of a 48-bit seconds part and a 32-bit nanoseconds part. The current time-of-day is derived by sampling the time-of-day seconds counter (SYS::PTP_TOD_LSB, SYS::PTP_TOD_MSB) and the nanoseconds counter (Tx time stamp) at the time of the frame’s departure from the switch. Note that the time-of-day value is adjusted for I/O path delay. For more information, see Section 2.2.1.8, Rx and Tx Time Stamps.

Set Reserved Bytes. This action write the frame’s Rx time stamp into the reserved bytes of the PTP header.

Clear Reserved Bytes. This action clears the reserved byte of the PTP header by setting the four bytes to 0.

In addition, the UDP checksum is handled for IPv4 frames and IPv6 frames after any PTP modifications by the rewriter. For IPv4, the UDP checksum is cleared while for IPv6, the two bytes immediately following the PTP header are updated so that the UDP checksum remains correct.

All internal calculations on nanoseconds time stamps are signed and use 48 bits of precision. When enabling a port for backplane mode, it is possible specify the number of bits to use from the 48-bit counter. This is specified in SYS::PTP_CFG.PTP_STAMP_WID. This means that the time stamp values carried in the reserved bytes or in the correction field in the PTP header can be for instance 30 bits instead of the default 32 bits.

2.13.5 SPECIAL REWRITER OPERATIONS

VCAP IS2 can trigger a special rewriter operation through IS2_ACTION.REW_OP[3:0] = 8 (rewriter special), where the frame's MAC addresses are swapped:

- 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.

In addition, VCAP IS2 can trigger replacement of the source MAC address by setting IS2_ACTION.SMAC_REPLACE_ENA = 1. The new source MAC address is configurable per egress port through SYS::REW_MAC_LOW_CFG and SYS::REW_MAC_HIGH_CFG.

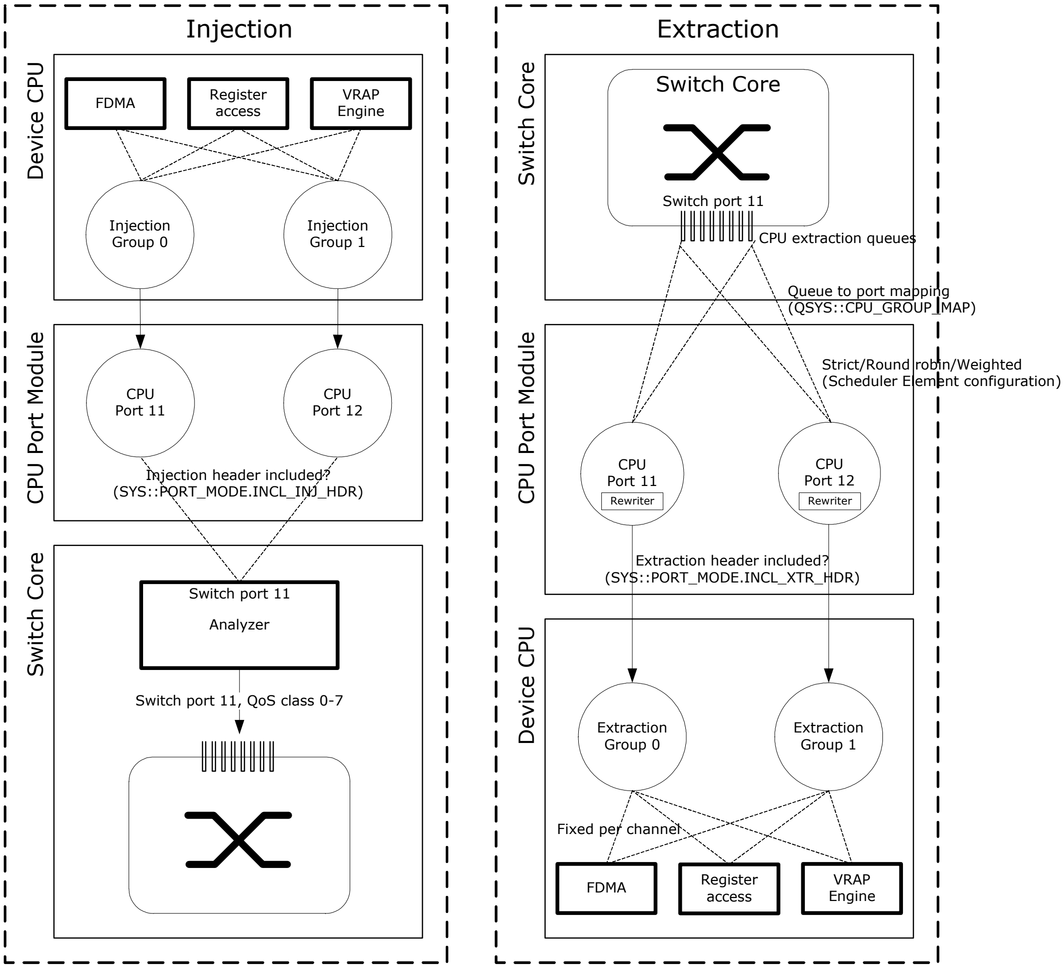
The operations of swapping of MAC addresses and replacing the source MAC address are mutually exclusive and must not be enabled at the same time.

2.14 CPU Port Module

The CPU port module connects the switch core to the CPU system so that frames can be injected from or extracted to the CPU. It is also possible to use a regular front port as a CPU port. This is known as a Node Processor Interface (NPI).

The following illustration shows how the switch core interfaces to the CPU system through the CPU port module for injection and extraction of frames.

FIGURE 2-38: CPU INJECTION AND EXTRACTION



2.14.1 FRAME EXTRACTION

The following table lists the registers associated with frame extraction.

TABLE 2-112: FRAME EXTRACTION REGISTERS

Register	Description	Replication
QSYS::CPU_GROUP_MAP	Configuration of mapping of extraction queues to CPU ports.	None
SYS::PORT_MODE.INCL_XTR_HDR	Enables insertion of extraction header. Configures formatting of outgoing frames.	Per CPU port (ports 11 and 12)

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 11 and port 12) 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 2.13, Rewriter.

The device CPU contains the functionality for reading out the frames. This can be done through the frame DMA or regular register access. The Versatile Register Access Protocol (VRAP) Engine can also be attached to one of the CPU extraction groups. This allows an external device to access internal registers through Ethernet frames.

The following table lists the contents of the CPU extraction header.

TABLE 2-113: CPU EXTRACTION HEADER

Field	Bit	Width	Description
RESERVED	127	1	Reserved.
REW_OP	117	9	Rewriter operation command. REW_OP[2:0] = 3 implies two-step PTP where REW_OP[8:3] contains the PTP time stamp identifier. Other settings do not apply.
REW_VAL	85	32	This field contains the Rx time when the frame was received.
LLEN	79	6	Frame length in bytes: 60 × WLEN + LLEN – 80.
WLEN	71	8	See LLEN.
RESERVED	47	24	Reserved.
SRC_PORT	43	4	The port number where the frame was received (0-11).
ACL_ID	37	6	If ACL_HIT is set, this value is the combined ACL_ID action of the rules hit in IS2.
RESERVED	36	1	Reserved.
SFLOW_ID	32	4	sFlow sampling ID. 0-11: Frame was sFlow sampled by a Tx sampler on port given by SFLOW_ID. 12: Frame was sFlow sampled by an Rx sampler on port given by SRC_PORT. 13-14: Reserved. 15: Frame was not sFlow sampled.
ACL_HIT	31	1	Set if frame has hit a rule in IS2, which copies the frame to the CPU (IS2 actions CPU_COPY_ENA or HIT_ME_ONCE).
DP	30	1	The frame's drop precedence (DP) level after policing.

TABLE 2-113: CPU EXTRACTION HEADER (CONTINUED)

Field	Bit	Width	Description
LRN_FLAGS	28	2	The source MAC address learning action triggered by the frame. 0: No learning. 1: Learning of a new entry. 2: Updating of an already learned unlocked entry. 3: Updating of an already learned locked entry.
CPUQ	20	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.
QOS_CLASS	17	3	The frame's classified QoS class.
TAG_TYPE	16	1	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.
PCP	13	3	The frame's classified PCP.
DEI	12	1	The frame's classified DEI.
VID	0	12	The frame's classified VID.

2.14.2 FRAME INJECTION

The following table lists the registers associated with frame injection.

TABLE 2-114: FRAME INJECTION REGISTERS

Register	Description	Replication
SYS::PORT_MODE.INCL_INJ_HDR	Enable parsing of injection header. Configures formatting of incoming frames.	Per CPU port (ports 11 and 12)
QSYS::EQ_PREFER_SRC	Enable preferred arbitration of the CPU port (port 11) over front ports.	CPU port (port 11 only)

The CPU injects frames through the two CPU injection groups that are independent of each other. The injection groups connect to the two CPU ports (port 11 and port 12) in the CPU port module. In 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 (port 11). This implies that both CPU injection groups consume memory resources from the shared queue system for port 11 and that analyzer configuration for port 11 are applied to all frames.

In the switch core, the CPU port can be preferred over other ingress ports when transferring frames to egress queues by enabling precedence of the CPU port (QSYS::EQ_PREFER_SRC).

The first 20 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 2.9.3, Forwarding Engine.

The contents of the CPU injection header is listed in the following table.

TABLE 2-115: CPU INJECTION HEADER

Field	Bit	Width	Description
BYPASS	127	1	When this bit is set, the analyzer processing is skipped for this frame. The destination set is specified in DEST and CPU_-QUEUE. Forwarding uses the QOS_CLASS, and the rewriter uses the tag information (POP_CNT, TAG_TYPE, PCP, DEI, VID) for rewriting actions. When this bit is cleared, the analyzer determines the destination set, QoS class, and VLAN classification for the frame through normal frame processing including lookups in the MAC table and VLAN table.
MASQ	126	1	When this bit is set, masquerading is enabled. The classifier, analyzer, and queue system handle the frame as if it was received by the ingress port specified in MASQ_PORT. This field overloads the REW_OP field and should only be used when BYPASS = 0.
MASQ_PORT	122	4	Masquerading port used when MASQ is set. This field overloads the REW_OP field and should only be used when BYPASS = 0.
REW_OP	126	1	If set, frame's source MAC address is replaced with source MAC address configured per egress port (SYS::REW_MAC_LOW_CFG, SYS::REW_MAC_HIGH_CFG).
REW_OP	117	9	Rewriter operation command. Used when BYPASS = 1. The following commands are supported: No operation: REW_OP[3:0] = 0. No operation. Special Rewrite: REW_OP[3:0] = 8. Swap the MAC addresses and clear bit 40 in the new SMAC when transmitting the frame. DSCP remark: REW_OP[2:0] = 1. REW_OP[8:3] contains the frame's classified DSCP to be used at egress for remarking. One-step PTP: REW_OP[2:0] = 2. The frame's residence time is added to the correction field in the PTP frame. The following sub-commands can be encoded: REW_OP[5]: Set if egress delay must be added to residence time. Two-step PTP: REW_OP[2:0] = 3. The frame's departure time stamp is saved in the time stamp FIFO queue at egress. REW_OP[8:3] contains the PTP time stamp identifier used by the time stamp FIFO queue. Identifiers 0 through 3 are pre-allocated to be used by CPU injected frames. Origin PTP: REW_OP[2:0] = 5. The time of day at the frame's departure time is written into the origintime stamp field in the PTP frame. Unspecified bits must be set to 0.
REW_VAL	85	32	By default, this field contains the frame's receive time stamp. For injected frames, this can be set by the CPU to indicate when the injection started. The rewriter can then calculate a residence time based on REW_VAL and the frame's transmission time stamp. If TFRM_TIMER > 0, then REW_VAL contains the transmission slot for periodic frame transmission (0 through 1023).
RESERVED	69	17	Reserved.

TABLE 2-115: CPU INJECTION HEADER (CONTINUED)

Field	Bit	Width	Description
DEST	56	12	This is the destination set for the frame. DEST[11] is the CPU. Used when BYPASS = 1.
RESERVED	47	9	Reserved.
SRC_PORT	43	4	The port number where the frame was injected (0-12).
RESERVED	41	2	Reserved.
TFRM_TIMER	37	4	Selects timer for periodic transmissions (1 through 8). If TFRM_-TIMER=0 then normal injection.
RESERVED	31	6	Reserved.
DP	30	1	The frame's drop precedence (DP) level after policing. Used when BYPASS = 1.
POP_CNT	28	2	Number of VLAN tags that must be popped in the rewriter before adding new tags. Used when BYPASS = 1. 0: No tags must be popped. 1: One tag must be popped. 2: Two tags must be popped. 3: Disable all rewriting of the frame. The rewriter can still update the FCS.
CPUQ	20	8	CPU extraction queue mask (one bit per CPU extraction queue). Each bit set implies the frame must be forwarded by the CPU to the specific queue. Used when BYPASS = 1 and DEST[11] = 1.
QOS_CLASS	17	3	The frame's classified QoS class. Used when BYPASS = 1.
TAG_TYPE	16	1	The tag information's associated Tag Protocol Identifier (TPID). Used when BYPASS = 1. 0: C-tag: EtherType = 0x8100. 1: S-tag: EtherType = 0x88A8 or custom value.
PCP	13	3	The frame's classified PCP. Used when BYPASS = 1.
DEI	12	1	The frame's classified DEI. Used when BYPASS = 1.
VID	0	12	The frame's classified VID. Used when BYPASS = 1.

2.14.3 NODE PROCESSOR INTERFACE (NPI)

The following table lists the registers associated with the NPI.

TABLE 2-116: NODE PROCESSOR INTERFACE REGISTERS

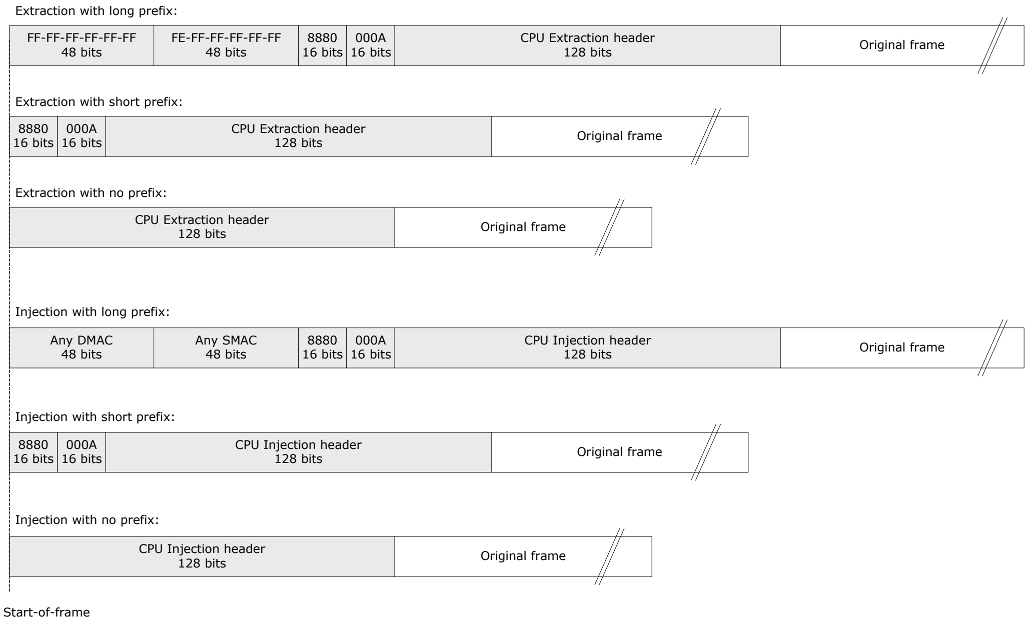
Register	Description	Replication
QSYS::EXT_CPU_CFG	Configuration of the NPI port number and configuration of which CPU extraction queues are redirected to the NPI.	None
SYS::PORT_MODE.INCL_X-TR_HDR	Enables insertion of extraction header.	Per port
SYS::PORT_MODE.INCL_INJ_HDR	Configuration of NPI ingress mode.	Per port

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.

A dual CPU system is possible where both the internal and the external CPU are active at the same time. Through QSYS::EXT_CPU_CFG.EXT_CPUQ_MSK, it is configurable to which of the eight CPU extraction queues are directed to the internal CPU and which are directed to external CPU. A frame can be extracted to both the internal CPU and the external CPU if the frame is extracted for multiple reasons.

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. The following illustration shows the different frame formats supported.

FIGURE 2-39: CPU INJECTION AND EXTRACTION PREFIXES


Extraction with long prefix:
FF-FF-FF-FF-FF-FF 48 bits
FE-FF-FF-FF-FF-FF 48 bits
8880 16 bits
000A 16 bits
CPU Extraction header 128 bits
Original frame
Extraction with short prefix:
8880 16 bits
000A 16 bits
CPU Extraction header 128 bits
Original frame
Extraction with no prefix:
CPU Extraction header 128 bits
Original frame
Injection with long prefix:
Any DMAC 48 bits
Any SMAC 48 bits
8880 16 bits
000A 16 bits
CPU Injection header 128 bits
Original frame
Injection with short prefix:
8880 16 bits
000A 16 bits
CPU Injection header 128 bits
Original frame
Injection with no prefix:
CPU Injection header 128 bits
Original frame
Start-of-frame

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 0x8880000A). 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 20 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 0x8880000A). 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.

2.14.4 FRAME GENERATION ENGINE FOR PERIODIC TRANSMISSIONS

The following table lists the registers associated with the frame generation engine.

TABLE 2-117: FRAME GENERATION ENGINE

Register	Description	Replication
QSYS::TFRM_MISC	Configuration to cancel ongoing periodic transmissions.	None
QSYS::TFRM_PORT_DLY_ENA	Enable spaced-out transmissions when multiple periodic transmissions are scheduled at the same time.	Per port

TABLE 2-117: FRAME GENERATION ENGINE (CONTINUED)

Register	Description	Replication
QSYS::TFRM_TIMER_CFG_x	Period configuration. These registers configure the transmission period between frames.	8

The device contains a frame generation engine that can periodically transmit frames on a port with a programmable transmission period. The transmission period can be as low as 200 ns, which implies wire-speed back-to-back transmissions, or as high as several minutes. Up to 1024 periodic transmissions can coexist with up to eight different transmission periods.

To set up a periodic transmission, the CPU must inject a setup frame using the following settings in the injection header:

- REW_VAL set to the selected transmission slot (0 through 1023).
- ACL_ID set to the selected timer (1 through 8). The transmission period is programmed in steps of 198.2 ns in TFRM_TIMER_CFG_x, where x = ACL_ID. Note that injecting a frame with ACL_ID > 0 effectively enables the periodic transmissions. To inject a normal frame, ACL_ID must be set to 0.
- BYPASS set to 1.
- DEST set to the Tx port to which the periodic transmission applies. Only one Tx port per transmission is possible.
- Other fields in the injection header are applicable the same way as for normal injections. Note, that the periodic transmissions are subject to normal rewriter operations before being transmitted on the Tx port. For more information, see Section 2.13, Rewriter.

After the injection, the setup frame is placed into the selected transmission slot and it is periodically transmitted using the transmission period defined by the selected timer. Every time the transmission period has passed since the last transmission, the frame is scheduled for a new transmission on the associated port. The periodic frame transmission takes precedence over other transmissions on the port, and as a consequence the frame is transmitted immediately after a potential ongoing transmission has ended.

If multiple periodic transmissions on a Tx port use the same timer, multiple frames are scheduled for transmission at the same time when the period has passed. By default, these frames are transmitted in a burst, back-to-back. However, it is also possible to space-out these frames over time and thereby allowing the normal data traffic to be interleaved the periodic transmissions. If the spacing is enabled (TFRM_PORT_DLY), timer 8 defines the period between frames in the burst.

Periodic transmissions are cancelled again by programming the slot number in TFRM_MISC.TIMED_CANCEL_SLOT and setting TFRM_MISC.TIMED_CANCEL_1SHOT. This removes the injected frame from the transmission slot.

Transmission slots 0 through 15 support any transmission periods including back-to-back transmissions while slots 16 through 1023 support transmission periods larger than 30 µs.

Frames transmitted from the frame generation engine are counted by the Tx counters just like normal frames using the QoS class specified in the injection header by the setup frame.

2.15 VRAP Engine

The Versatile Register Access Protocol (VRAP) engine allows external equipment to access registers in the device through any of the Ethernet port's on the device. 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 registers listed in the following table control the VRAP engine.

TABLE 2-118: VRAP REGISTERS

Target:Register_group:Register.field	Description	Replication
ANA:CPU_FWD_CFG:CPU_VRAP_REDIR_ENA	Enable redirection of VRAP frames to CPU.	Per port
ANA:CPUQ_CFG2:CPUQ_VRAP	Configure the CPU extraction queue for VRAP frames.	None

TABLE 2-118: VRAP REGISTERS (CONTINUED)

Target:Register_group:Register.field	Description	Replication
QSYS:CPU_GROUP_MAP:CPU_GROUP_MAP	Map VRAP CPU extraction queue to a CPU port. One CPU port must be reserved for VRAP frames.	None
DEVCPU_QS:XTR_GRP_CFG:MODE	Enable VRAP mode for reserved CPU port.	Per CPU port
DEVCPU_QS:INJ_GRP_CFG:MODE	Enable VRAP mode injection for reserved CPU port.	Per CPU port
SYS:PORT_MODE:INCL_INJ_HDR	The injection header is not present for VRAP response frames.	Per port
SYS:PORT_MODE:INCL_XTR_HDR	The extraction header is not present for redirected VRAP frames.	Per port
DEVCPU_GCB:VRAP_ACCESS_STAT	VRAP access status.	None

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 2.7.6, 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 and DEVCPU_QS:INJ_GRP: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).

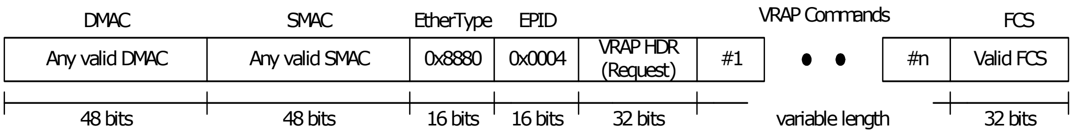
The complete VRAP functionality can be enabled automatically at chip startup by the use of special chip boot modes. For more information, see Section 3.1, VCore-III Configurations.

The following describes the VRAP frame formats.

2.15.1 VRAP REQUEST FRAME FORMAT

The following illustration shows the format of a VRAP request frame.

FIGURE 2-40: 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 but is useful for padding and identification purposes.
- PAUSE: Does not access registers but causes the VRAP engine to pause between register access.

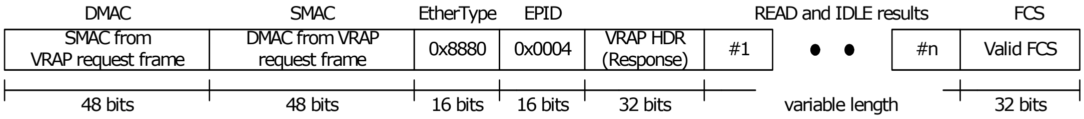
Each of the VRAP commands are described in the following sections. Each VRAP request frame can contain multiple VRAP commands. Commands are processed sequentially starting with VRAP command #1, #2, and so on. For more information, see <figure reference>. There are no restrictions on the order or number of commands in the frame.

2.15.2 VRAP RESPONSE FRAME FORMAT

Having processed a VRAP request frame by executing the register access commands contained in the frame, the VRAP engine generates a VRAP response frame. The response frame uses the swapped DMAC and SMAC addresses in the VRAP request frame and it is transmitted on port where the VRAP request frame was received. If the DMAC in the VRAP request frame is a multicast DMAC, then bit 40 is cleared before using the DMAC as the SMAC in the VRAP response frame. The results of the VRAP commands are inserted into the response frame in the same order as the VRAP commands were processed.

The following illustration shows the format of a VRAP response frame.

FIGURE 2-41: VRAP RESPONSE FRAME FORMAT



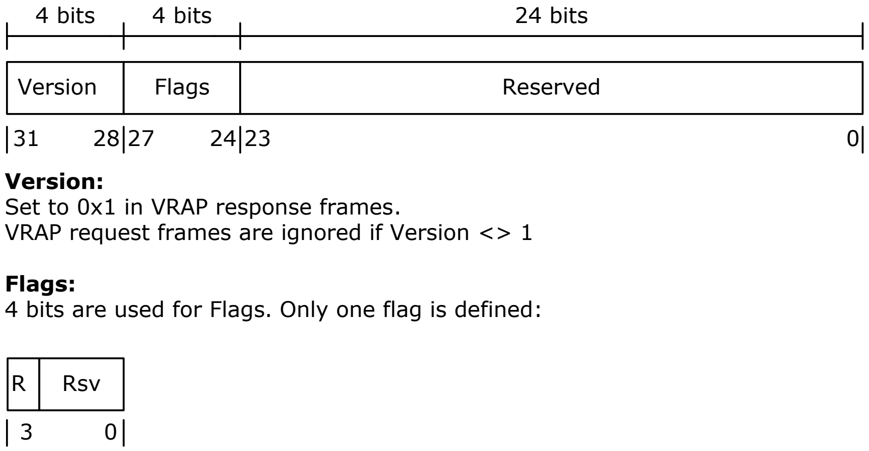
The VRAP response frame follows the VRAP request frame in terms of VLAN tagging: If the VRAP request was tagged, the VRAP response is also tagged using the same VLAN.

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.

2.15.3 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. The following illustration shows the VRAP header.

FIGURE 2-42: 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.

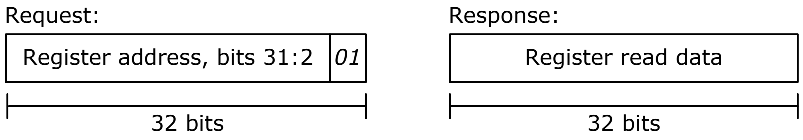
A valid VRAP request frame must use Version = 1 and Flags.R = 1. Frames with an invalid VRAP header are discarded and not processed by the VRAP engine.

2.15.4 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. The following illustration shows the request command and the associated response result.

FIGURE 2-43: READ COMMAND

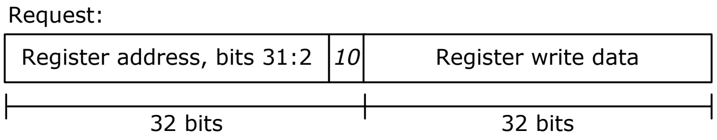


2.15.5 VRAP WRITE COMMAND

The WRITE command writes a 32-bit value to any register inside the device.

The WRITE command is 8 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 10, followed by one 32-bit write-data field. The following illustration shows the command.

FIGURE 2-44: WRITE COMMAND

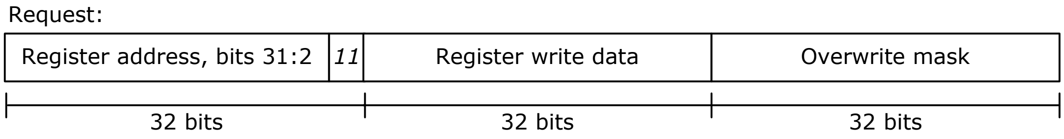


2.15.6 VRAP READ-MODIFY-WRITE COMMAND

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. The following figure shows the command.

FIGURE 2-45: READ-MODIFY-WRITE COMMAND

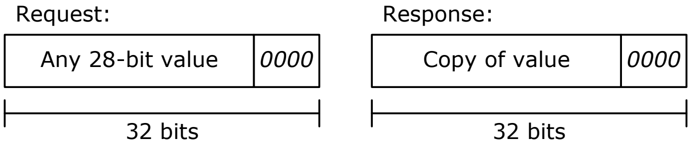


2.15.7 VRAP IDLE COMMAND

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. The following illustration depicts the request command and the associated response.

FIGURE 2-46: IDLE COMMAND

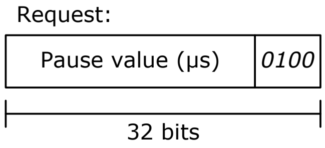


2.15.8 VRAP PAUSE COMMAND

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. The following illustration depicts the PAUSE command.

FIGURE 2-47: PAUSE COMMAND



2.16 Layer 1 Timing

There are eight recovered clocks, four outputs that provide timing sources for external timing circuitry in redundant timing implementations, and four internal clocks for the timing-recovery circuit. The following tables list the registers and pins associated with Layer 1 timing.

TABLE 2-119: LAYER 1 TIMING CONFIGURATION REGISTERS

Register	Description
HSIO::SYNC_ETH_CFG	Configures recovered clocks. Replicated per recovered clock.
HSIO::SYNC_ETH_PLL_CFG	Additional PLL recovered clock configuration. Replicated per recovered clock.
HSIO::PLL5G_CFG3	Enables high speed clock output.
PHY:PHY_GP:PHY_RCVD_CLK0_CTRL	Configures PHY recovered clock 0.
PHY:PHY_GP:PHY_RCVD_CLK1_CTRL	Configures PHY recovered clock 1.

TABLE 2-120: LAYER 1 TIMING RECOVERED CLOCK PINS

Pin Name	I/O	Description
RCVRD_CLK0	O	Recovered clock output, configured by SYNC_ETH_CFG[0]. This is an overlaid function on GPIO.
RCVRD_CLK1	O	Recovered clock output, configured by SYNC_ETH_CFG[1]. This is an overlaid function on GPIO.
CLKOUT	O	PLL high speed clock output.

It is possible to recover receive timing from any 10/100/1000 Mbps and 2.5 Gbps data streams into the device.

The recovered clock outputs have individual divider configuration (through SYNC_ETH_CFG.SEL_RCVRD_CLK_DIV) to allow division of SerDes receive frequency by 1, 2, 4, 5, 8, 16, or 25.

The recovered clocks are single-ended outputs, and the suggested divider settings in the following tables are selected to make sure to not output too high a frequency through the input and outputs.

The four CuPHY ports share two recovered clocks. In table Table 2-121, the two clock resources are called clk and can take the value 0 or 1. The CLK_SRC_SEL0 and CLK_SRC_SEL1 fields are programmed with the number of the CuPHY that provides the recovered clock; CuPHYcan take the value 0 through 3.

TABLE 2-121: RECOVERED CLOCK SETTINGS FOR 1 GBPS AND LOWER

Interface	Output Frequency	Register Settings for Output *n*
CuPHY clk 0-1	31.25 MHz	SYNC_ETH_CFG[n].RCVRD_CLK_ENA=1, SYNC_ETH_CFG[n].SEL_RCVRD_CLK_DIV=1, SYNC_ETH_CFG[n].SEL_RCVRD_CLK_SRC=clk, PHY_RCVD_CLK[clk]_CTRL.RCVD_CLK[clk]_ENA=1, PHY_RCVD_CLK[clk]_CTRL.CLK_SRC_SEL[clk]=(CuPHY), PHY_RCVD_CLK[clk]_CTRL.CLK_FREQ_SEL[clk]=1, and PHY_RCVD_CLK[clk]_CTRL.CLK_SEL_PHY[clk]=1
SerDes 0-8	31.25 MHz	SYNC_ETH_CFG[n].RCVRD_CLK_ENA=1, SYNC_ETH_CFG[n].SEL_RCVRD_CLK_DIV=1, and SYNC_ETH_CFG[n].SEL_RCVRD_CLK_SRC=(SerDes+2)

TABLE 2-122: RECOVERED CLOCK SETTINGS FOR 2.5 GBPS

Interface	Output Frequency	Register Settings for Output *n*
SerDes 7-8	31.25 MHz	SYNC_ETH_CFG[n].RCVRD_CLK_ENA=1, SYNC_ETH_CFG[n].SEL_RCVRD_CLK_DIV=4, and SYNC_ETH_CFG[n].SEL_RCVRD_CLK_SRC=(Serdes + 2)

The frequency of the PLL can be used as recovered clock. The following table shows the configurations.

TABLE 2-123: RECOVERED CLOCK SETTINGS FOR PLL

PLL	Output Frequency	Register Settings for Output *n*
PLL	31.25 MHz	SYNC_ETH_CFG[n].RCVRD_CLK_ENA=1, SYNC_ETH_CFG[n].SEL_RCVRD_CLK_DIV=1, and SYNC_ETH_CFG[n].SEL_RCVRD_CLK_SRC=11

The recovered clock from the PLL can also be sent directly out on the differential high-speed CLKOUT output. The recovered clock frequency must be set to copy the switch core using HSIO::PLL5G_CFG3.CLKOUT_SEL.

It is possible to automatically squelch the clock output when the device detects a loss of signal on an incoming data stream. This can be used for failover in external timing recovery solutions.

The following table lists how to configure squelch for possible recovered clock sources (configured in SYNC_ETH_CFG[n].SEL_RCVRD_CLK_SRC).

TABLE 2-124: SQUELCH CONFIGURATION FOR SOURCES

SRC	Associated Squelch Configuration
0-5	Set SERDES1G_COMMON_CFG.SE_AUTO_SQUELCH_ENA in SD macro to enable squelch when receive signal is lost. SD1G macro index is (SRC).
6-8	Set SERDES6G_COMMON_CFG.SE_AUTO_SQUELCH_ENA in SD macro to enable squelch when receive signal is lost. SD6G macro-index is (SRC-6).

When squelching the clock, it stops when it detects loss of signal (or PLL lock). The clock stops on either high or low level.

The auto squelch function is not supported when 100 Mbit operation on SerDes, so in this mode the auto squelch function must be disabled.

2.17 Hardware Time Stamping

Hardware time stamping provides nanosecond-accurate frame arrival and departure time stamps, which are used to obtain high precision timing synchronization and timing distribution.

All frames are Rx time stamped on arrival with a 48-bit time stamp value using a hardware timer (time stamper) that is implemented in the Media Access Control (MAC) block. The Rx time stamper provides high time stamp accuracy relative to actual arrival time of the first byte of the frame from the PHY device. Within the VCAP IS2, it is decided if the frame and associated Rx time stamp must be redirected or copied to CPU for processing. The frame is forwarded as normal otherwise.

The VCAP IS2 also decides if a Tx time stamp must be triggered for a frame. Given the Rx and Tx time stamps, the frame's residence time inside the switch is calculated. The residence time can be stored in a time stamp queue for the CPU to access (two-step time stamping) or the residence time can be used to update the residence time field inside Precision Time Protocol frames (one-step time stamping).

The Tx time stamper is located at the transmit side of the MAC block as close to the PHY device as possible and provides high accuracy of time stamp relative to when the first byte of the frame is actually transmitted to the PHY.

The device also implements a time of day counter with nanosecond-accuracy. The time of day counter is derived from a one-second timer. The one-second timer generates a pulse per second and is either derived from an adjusted system clock or from external timing equipment.

2.17.1 TIME STAMP CLASSIFICATION

Frames requiring Rx or Tx time stamping are identified by VCAP IS2. The IS2 action that triggers time stamping is REW_OP[2:0], with the following options:

- One-step time stamping (REW_OP[2:0] = 2), which implies adding the frame's residence time to the correction field.
- Two-step time stamping (REW_OP[2:0] = 3), which implies saving the frame's Tx time in a time stamp queue.
- Origin time stamping (REW_OP[2:0] = 5), which implies writing the time of day into the origintime stamp field.

IS2 can be configured to identify the following frame formats from IEEE 1588-2008:

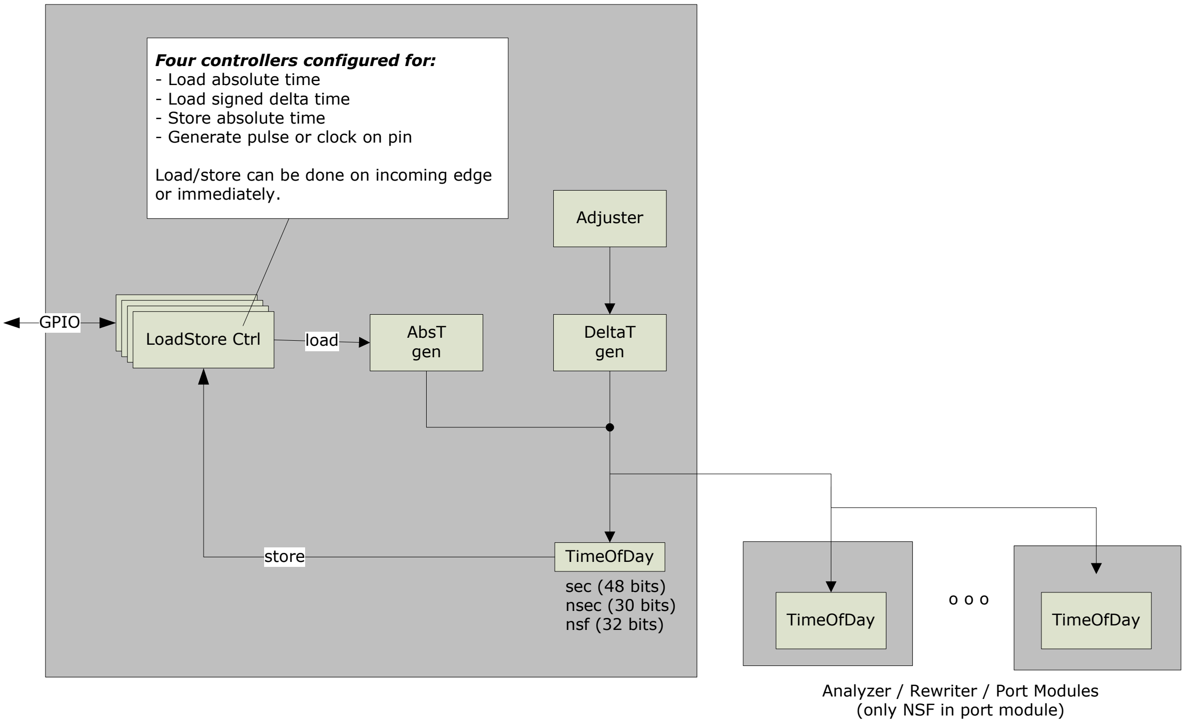
- Transport of PTP over UDP over IPv4
- Transport of PTP over UDP over IPv6
- Transport of PTP over IEEE 802.3/Ethernet

In addition, frames can be encapsulated in Q-in-Q tunnels. For more information about the frame encapsulations and PTP protocol fields supported by the VSC7418-01 device, see Section 2.8.3, VCAP IS2.

2.17.2 TIME OF DAY GENERATION

The DEVCPU has connection to four GPIOs to be used for 1588 synchronization shown in the following illustration.

FIGURE 2-48: TIMING DISTRIBUTION



Each block using timing has a TimeOfDay instance included. These modules let time pass according to an incoming delta request, instructing it to add the nominal clock period in nanoseconds (+0/+1/–1) for each system clock cycle. This is controlled by a deltaT function in the DEVCPU, which can be configured to do regular adjustments to the time by adding a single nanosecond more or less at specified intervals. The absT function in the DEVCPU can set the full TOD time in the system. This is controlled by the LoadStore controllers.

The DEVCPU includes four LoadStore controllers, each using a designated GPIO pin on the GPIO interface. LoadStore controller 0 uses PTP_0 pin; LoadStore controller 1 uses PTP_1 pin, and so forth. Before using the LoadStore controller, the VCore-III CPU must enable the overlaid functions for the appropriate GPIO pins. For more information, see Section 3.8.8.1, GPIO Overlaid Functions.

Each controller has CPU accessible registers with the full time of day set, and can be configured to load these into the TimeOfDay instances, or to store the current values from them. The operation can be done on a detected edge on the associated pin or immediately. The GPIO pin can also generate a clock with configurable high and low periods set in nanoseconds using the TimeOfDay watches or it can generate a pulse at a configured time with a configurable duration and polarity.

TABLE 2-125: LOADSTORE CONTROLLER

Pin Control Field	Function
Action	Load: Load TOD_SEC and TOD_NSEC through the absTime bus. Delta: Add configured nsec to the current time of day (absT). Store: Store the current TOD_SEC and TOD_NSEC. Clock: Generate a clock or a pulse on the pin.
Sync	Execute the load/store on incoming edge, if action is LOAD or STORE. Generate a pulse instead of clock if action is CLOCK.
Inverse polarity	Falling edges are detected/generated.
TOD_sec	The 48-bit seconds of a time of day set to be loaded or stored.
TOD_nsec	The 30-bit nanoseconds of a time of day set to be loaded or stored.
Waveform high	Number of nanoseconds in the high period for generated clocks. Duration of pulse for generated pulse.

TABLE 2-125: LOADSTORE CONTROLLER (CONTINUED)

Pin Control Field	Function
Waveform_low	Number of nanoseconds in the low period for generated clocks. Delay from TOD_ns = 0 for generated pulse.

In addition, the load operation can load a delta to the current time. This is done by executing a LOAD action but using the DELTA command.

Each operation generates an interrupt when executed. For the clock action, the interrupt is generated when the output switches to the active level. The interrupts from each controller can be masked and monitored by the interrupt controller in the ICPU_CFG register target.

The four controllers are completely equal in their capabilities.

2.17.3 HARDWARE TIME STAMPING MODULE

This section explains the functions of the hardware time stamping module. The following table lists the registers associated with the hardware time stamping module.

TABLE 2-126: HARDWARE TIME STAMPING REGISTERS

Register	Description	Replication
SYS::PTP_TXSTAMP	time stamp value in time stamp queue.	None
SYS::PTP_NXT	Advancing the time stamp queue.	None
SYS::PTP_STATUS	time stamp queue status and entry data.	None
ANA::PTP_ID_HIGH	Release of time stamp identifiers, values 32 through 63.	None
ANA::PTP_ID_LOW	Release of time stamp identifiers, values 0 through 31.	None

Each port module contains a hardware time stamping module that measures arrival and departure times based on the master timer. For information about how the Rx and Tx time stamps are derived, see Section 2.2.1.8, Rx and Tx Time Stamps.

2.17.3.1 Two-Step Time Stamping

Two-step time stamping is performed if the IS2 rewriter action is two-step (IS2_ACTION.REW_OP[2:0] = 3). This action can be applied to any frame, also non-PTP frames, because the frame itself is not modified. The frame’s Tx time stamp is stored in a time stamp FIFO queue, which the CPU can access (SYS::PTP_STATUS). The time stamp is common for all egress ports and can contain up to 128 time stamps. Each entry in the time stamp queue contains the following fields:

- SYS::PTP_STATUS.PTP_MESS_VLD: A 1-bit valid bit meaning the entry is ready for reading.
- SYS::PTP_STATUS.PTP_MESS_ID: A 6-bit time stamp identifier. A unique time stamp identifier is assigned to each frame for which one or more Tx time stamps are generated. The time stamp identifier is also available in the CPU extraction header for frames extracted to the CPU. The time stamp identifier overloads the DSCP value in the CPU extraction header. For more information about the CPU extraction header, see Table 2-113. By providing the time stamp identifier in both the time stamp queue and in the extracted frames, the CPU can correlate which time stamps belong to which frames. Note that time stamp identifier value 63 implies that no free identifier could be assigned to the frame. The time stamp entry can therefore not be trusted.
- SYS::PTP_STATUS.PTP_MESS_TXPORT: The port number where the frame is transmitted. When transmitting a frame on multiple ports, there are generated multiple entries in the time stamp queue. Each entry uses the same time stamp identifier but with different Tx port numbers.
- SYS::PTP_STATUS.PTP_TXSTAMP_OAM: For two-step time stamping, this field is always 0. If set, it identifies an entry inserted by the Vitesse OAM processor (VOP).
- SYS::PTP_TXSTAMP: The frame’s Tx time stamp.

The time stamp queue is a simple FIFO that can be read by the CPU. The time stamp queue provides the following handles for reading:

- Overflow of the queue is signaled through SYS::PTP_STATUS.PTP_OVFL. Overflow implies that one or more time stamps could not be enqueued due to all 128 entries being in use. time stamps not enqueued are lost.
- The head-of-line entry is read through SYS::PTP_STATUS and SYS::PTP_TXSTAMP.
- Writing to the one-shot register SYS::PTP_NXT, removes the current head-of-line entry and advances the pointer

to the next entry in the time stamp queue.

When two-step Tx time stamping is performed for a frame destined for the CPU extraction queues, no entry in the time stamp FIFO queue is made. The frame's Rx time stamp is available through the CPU extraction header.

The time stamp identifiers can take values between 0 to 63. Value 63 implies that all values 0-62 are in use. Values 0 – 3 are pre-assigned to the CPU to be used for injection of frames. The remaining values are assigned by the analyzer to frames requesting time stamping through the VCAP IS2 action. The assigned values must be released again by the CPU by writing to the corresponding bit in ANA::PTP_ID_HIGH (values 32 through 63) or ANA::PTP_ID_LOW (values 0 through 31). The CPU releases a time stamp identifier when it has read the anticipated time stamp entries from the time stamp queue. Note that multicasted frames generate a time stamp entry per egress port using the same time stamp identifier. Each of these entries must be read before the time stamp identifier is released.

Two-step time stamping can be disabled per egress port using REW:PORT:PTP_CFG.PTP_2STEP_DIS. This setting overrules the IS2 action.

2.17.4 CONFIGURING I/O DELAYS

After a valid link is established and detected by the involved PCS logic, the I/O delays from the internal time stamping points to the serial line must be configured. The delays are both mode-specific and interface-specific, depending on the core clock frequency.

Ingress barrel shifter states that in 1G mode, the Rx delays must be added 0.8 ns times the value of PCS1G_LINK_STATUS.DELAY_VAR to adjust for barrel shifting state after link establishment. In 2.5G mode, the multiplier is 0.32 ns. In 100FX mode, the Rx delays must be subtracted 0.8 ns times the value of PCS_FX100_STATUS.EDGE_POS_PTP to adjust for detected data phase.

The Rx and Tx delay values for the different ports and modes are automatically configured in the software API.

2.18 Clocking and Reset

The reference clock for the PLL (REFCLK_P/N) is either differential or single-ended. The frequency can be 25 MHz, 125 MHz, 156.25 MHz, or 250 MHz. The PLL must be configured for the appropriate clock frequency by using strapping inputs REFCLK_CONF[2:0].

The PLL can be used as a recovered clock source for Synchronous Ethernet. For information about how to configure PLL clock recovery, see Section 2.16, Layer 1 Timing.

For information about protecting the VCore CPU system during a soft-reset, see Section 2.18, Clocking and Reset.

2.18.1 PIN STRAPPING

Configure PLL reference clock and VCore startup mode using the strapping pins on the GPIO interface. The device latches strapping pins and keeps their value when nRESET to the device is released. After reset is released, the strapping pins are used for other functions. For more information about which GPIO pins are used for strapping, see Section 3.8.8.1, GPIO Overlaid Functions.

TABLE 2-127: STRAPPING

Pin	Description
REFCLK_CONF[2:0]	Configuration of reference clock frequency for PLL. 000: 125 MHz 001: 156.25 MHz 010: 250 MHz 100: 25 MHz Other values are reserved and must not be used.

TABLE 2-127: STRAPPING (CONTINUED)

Pin	Description
VCORE_CFG[3:0]	Configuration of VCore system startup conditions. 0000: VCore-III CPU is enabled (Little Endian mode) and boots from SI (the SI slave is disabled). 1001: PCIe 1.x endpoint is enabled. Automatic boot of VCore-III CPU is disabled, and SI slave is enabled. 1010: MIIM slave is enabled with MIIM address 0 (MIIM slave pins are overlaid on GPIOs). Automatic boot of VCore-III CPU is disabled, and SI slave is enabled. 1011: MIIM slave is enabled with MIIM address 31 (MIIM slave pins are overlaid on GPIOs). Automatic boot of VCore-III CPU is disabled, and SI slave is enabled. 1100: VCore-III CPU is enabled (Big Endian mode) and boots from SI (the SI slave is disabled). 1111: Automatic boot of VCore-III CPU is disabled, and SI slave is enabled. Other values are reserved and must not be used.

By using resistors to pull the GPIOs either low or high, software can use these GPIO pins for other functions after reset has been released. For more information about overlaid functions on the GPIOs, see Section 3.8.8.1, GPIO Overlaid Functions.

Undefined configurations are reserved and cannot be used. VCore-III configurations that enable a front port or the PCIe endpoint drive VCore_CFG[3:2] high when the front port or PCIe endpoint is ready to use.

3.0 VCORE-III SYSTEM AND CPU INTERFACES

This section provides information about the functional aspects of blocks and interfaces related to the VCore-III on-chip microprocessor system and an external CPU system.

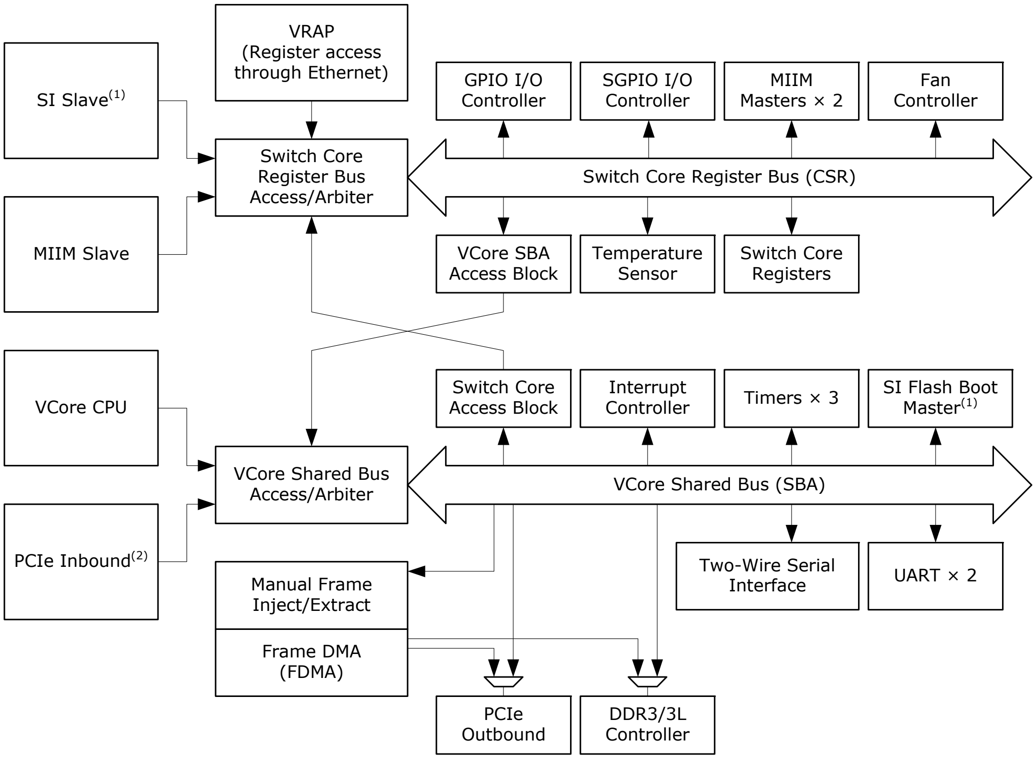
The device contains a powerful VCore-III CPU system that is based on an embedded MIPS24KEc-compatible microprocessor and a high bandwidth Ethernet Frame DMA engine. The VCore-III system can control the device independently, or it can support an external CPU, relieving the external CPU of the otherwise time consuming tasks of transferring frames, maintaining the switch core, and handling networking protocols.

When the VCore-III CPU is enabled, it either automatically boots up from serial Flash or an external CPU can manually load a code-image to the device and then start the VCore-III CPU.

An external CPU can be connected to the device through the PCIe interface, serial interface (SI), dedicated MIIM slave interface, or through Versatile Register Access Protocol (VRAP) formatted Ethernet frames using the NPI Ethernet port. Through these interfaces, the external CPU has access to the complete set of device registers and can control them without help from the VCore-IeVCore-III CPU if needed.

The following illustration shows the VCore-III block diagram.

FIGURE 3-1: VCORE-III SYSTEM BLOCK DIAGRAM.



1. When the Vcore-III CPU boots up from the SI Flash, the SI is reserved as boot interface and cannot be used by an external CPU.

2. Inbound PCIe access to BAR0 maps to VCore register space (switch core access block, interrupt controller, timers, two-wire serial master/slave, UARTs, and manual frame injection/extraction). Inbound PCIe access to BAR2 maps to the DDR3/3L memory space (DDR controller).

3.1 VCore-III Configurations

The behavior of the VCore-III system after a reset is determined by four VCore strapping pins that are overlaid on GPIO pins. The value of these GPIOs is sampled shortly after releasing reset to the device. For more information about the strapping pins, see Section 2.18.1, Pin Strapping.

The strapping value determines the reset value for the ICPU_CFG::GENERAL_CTRL register. After startup, the behavior of the VCore-III system can be modified by changing some of the fields in this register. The following are common scenarios.

- After starting the device with the VCore-III CPU disabled, an external CPU can manually boot the VCore-III CPU from SI Flash by writing ICPU_CFG::GENERAL_CTRL.IF_SI_OWNER = 1, ICPU_CFG::GENERAL_CTRL.BOOT_MODE_ENA = 1, and ICPU_CFG::GENERAL_CTRL.CPU_DIS = 0. Endianess is configured by ICPU_CFG::GENERAL_CTRL.CPU_BE_ENA. Setting GENERAL_CTRL.IF_SI_OWNER disables the SI slave, so the external CPU must use another interface than SI.

- After starting the device with the VCore-III CPU disabled and loading software into DDR3/DDR3Lmemory, an external CPU can boot the VCore-III CPU from DDR3/DDR3L memory by writing ICPU_CFG::GENERAL_CTRL.BOOT_MODE_ENA = 0 and ICPU_CFG::GENERAL_CTRL.CPU_DIS = 0. Endianess is configured using ICPU_CFG::GENERAL_CTRL.CPU_BE_ENA.
- After automatically booting from SI Flash, the VCore-III CPU can release the SI interface and enable the SI slave by writing ICPU_CFG::GENERAL_CTRL.IF_SI_OWNER = 0. This enables SI access from an external CPU to the device. A special PCB design is required to make the serial interface work for both Flash and external CPU access.
- MIIM slave can be manually enabled by writing ICPU_CFG::GENERAL_CTRL.IF_MIIM_SLV_ENA = 1. The MIIM slave automatically takes control of the appropriate GPIO pins.

The EJTAG interface of the VCore-III CPU and the Boundary Scan JTAG controller are both multiplexed onto the JTAG interface of the device. When the JTAG_ICE_nEN pin is low, the MIPS's EJTAG controller is selected. When the JTAG_ICE_nEN pin is high, the Boundary Scan JTAG controller is selected.

3.2 Clocking and Reset

The following table lists the registers associated with reset and the watchdog timer.

TABLE 3-1: CLOCKING AND RESET CONFIGURATION REGISTERS

Register	Description
ICPU_CFG::RESET	VCore-III reset protection scheme and initiating soft reset of the VCore-III system and/or CPU.
DEVCPU_GCB::SOFT_RST	Initiating chip-level soft reset.
ICPU_CFG::WDT	Watchdog timer configuration and status.

The VCore-III CPU runs 500 MHz, the DDR3/DDR3L controller runs 312.50 MHz, and the rest of the VCore-III system runs at 250 MHz.

The VCore-III can be soft reset by setting RESET.CORE_RST_FORCE. By default, this resets both the VCore-III CPU and the VCore-III system. The VCore-III system can be excluded from a soft reset by setting RESET.CORE_RST_CPU_ONLY; soft reset using CORE_RST_FORCE only then resets the VCore-III CPU. The frame DMA must be disabled prior to a soft reset of the VCore-III system. When CORE_RST_CPU_ONLY is set, the frame DMA, PCIe endpoint, and DDR3/DDR3L controller are not affected by a soft reset and continue to operate throughout soft reset of the VCore-III CPU.

The VCore-III system comprises all the blocks attached to the VCore Shared Bus (SBA), including the PCIe, DDR3/ DDR3L, and frame DMA/injection/extraction blocks. Blocks attached to the switch core Register Bus (CSR), including VRAP, SI, and MIIM slaves, are not part of the VCore-III system reset domain. For more information about the VCore-III system blocks, see Figure 3.1,.

The device can be soft reset by writing SOFT_RST.SOFT_CHIP_RST. The VCore-III system and CPU can be protected from a device soft reset by writing RESET.CORE_RST_PROTECT = 1 before initiating a soft reset. In this case, a chip-level soft reset is applied to all other blocks, except the VCore-III system and the CPU. When protecting the VCore-III system and CPU from a soft reset, the frame DMA must be disabled prior to a chip-level soft reset. The SERDES and PLL blocks can be protected from reset by writing to SOFT_RST.SOFT_SWC_RST instead of SOFT_CHIP_RST.

The VCore-III general purpose registers (ICPU_CFG::GPR) and GPIO alternate modes (DEVCPU_GCB::GPIO_ALT) are not affected by a soft reset. These registers are only reset when an external reset is asserted.

3.2.1 WATCHDOG TIMER

The VCore system has a built-in watchdog timer (WDT) with a two-second timeout cycle. The watchdog timer is enabled, disabled, or reset through the WDT register. The watchdog timer is disabled by default.

After the watchdog timer is enabled, it must be regularly reset by software. Otherwise, it times out and cause a VCore soft reset equivalent to setting RESET.CORE_RST_FORCE. Improper use of the WDT.WDT_LOCK causes an immediate timeout-reset as if the watchdog timer had timed out. The WDT.WDT_STATUS field shows if the last VCore-III CPU reset was caused by WDT timeout or regular reset (possibly soft reset). The WDT.WDT_STATUS field is updated only during VCore-III CPU reset.

To enable or to reset the watchdog timer, write the locking sequence, as described in WDT.WDT_LOCK, at the same time as setting the WDT.WDT_ENABLE field.

Note: Because watchdog timeout is equivalent to setting RESET.CORE_RST_FORCE, the RESET.CORE_RST_CPU_ONLY field also applies to watchdog initiated soft reset.

3.3 Shared Bus

The shared bus is a 32-bit address and 32-bit data bus with dedicated master and slave interfaces that interconnect all the blocks in the VCore-III system. The VCore-III CPU, PCIe inbound, and VCore SBA access block are masters on the shared bus; only they can start access on the bus.

The shared bus uses byte addresses, and transfers of 8, 16, or 32 bits can be made. To increase performance, bursting of multiple 32-bit words on the shared bus can be performed.

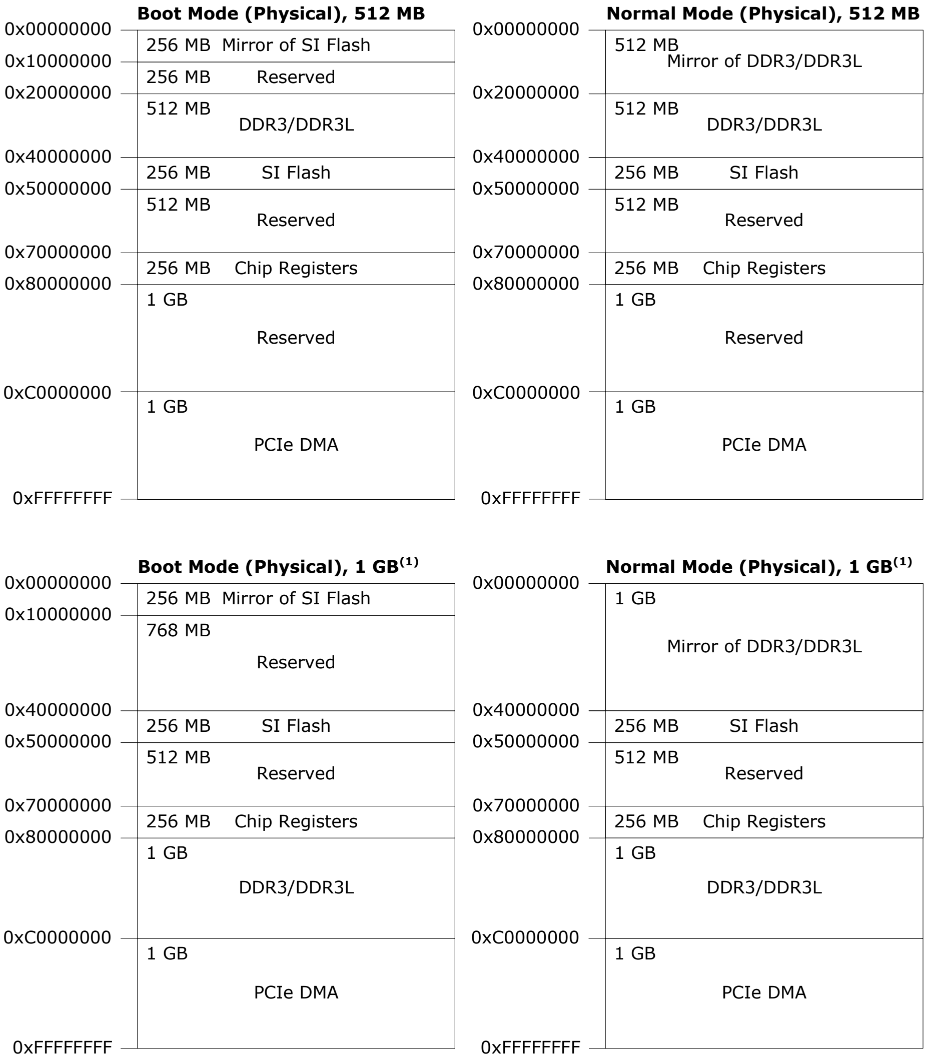
All slaves are mapped into the VCore-III system’s 32-bit address space and can be accessed directly by masters on the shared bus. There are two possible mappings of VCore-III shared bus slaves, boot mode and normal mode.

- Boot mode is active after power-up and reset of the VCore-III system. In this mode, the SI Flash boot master is mirrored into the lowest address region.
- In normal mode, the DDR3/DDR3L controller is mirrored into the lowest address region.

Changing between boot mode and normal mode is done by first writing and then reading ICPU_CFG::GENERAL_CTRL.BOOT_MODE_ENA. A change takes effect during the read.

The device supports up to 1 gigabyte (GB) of DDR3/DDR3L memory. By default, only 512 megabytes (MB) is accessible in the memory map. Write ICPU_CFG::MEMCTRL_CFG.512MBYTE_PLUS = 1 to accommodate more than 512 MB of memory.

FIGURE 3-2: SHARED BUS MEMORY



1. To enable support 1 gigabyte (GB) of DDR3/DDR3L memory (and the associated memory map), set ICPU_CFG::MEMCTRL_CFG.DDR_512MBYTE_PLUS.

Note: When the VCore-III system is protected from a soft reset using ICPU_CFG::RESET.CORE_RST_CPU_ONLY, a soft reset does not change shared bus memory mapping. For more information about protecting the VCore-III system when using a soft reset, see Figure 3.2,.

If the boot process copies the SI Flash image to DDR3/DDR3L, and if the contents of the SI memory and the DDR3 memory are the same, software can execute from the mirrored region when swapping from boot mode to normal mode. Otherwise, software must be execute from the fixed SI Flash region when changing from boot mode to normal mode.

The Frame DMA has dedicated access to PCIe outbound and to the DDR3/DDR3L. This means that access on the SBA to other parts of the device, such as register access, does not affect Frame DMA injection/extraction performance.

3.3.1 VCORE-III SHARED BUS ARBITRATION

The following table lists the registers associated with the shared bus arbitration.

TABLE 3-2: SHARED BUS CONFIGURATION REGISTERS

Register	Description
SBA::PL_CPU	Master priorities
SBA::PL_PCIE	Master priorities

TABLE 3-2: SHARED BUS CONFIGURATION REGISTERS (CONTINUED)

Register	Description
SBA::PL_CSR	Master priorities
SBA::WT_EN	Enable of weighted token scheme
SBA::WT_TCL	Weighted token refresh period
SBA::WT_CPU	Token weights for weighted token scheme
SBA::WT_PCIE	Token weights for weighted token scheme
SBA::WT_CSR	Token weights for weighted token scheme

The VCore-III shared bus arbitrates between masters that want to access the bus. The default is to use a strict prioritized arbitration scheme where the VCore-III CPU has highest priority. The strict priorities can be changed using registers PL_CPU, PL_PCIE, and PL_CSR.

- *_CPU registers apply to VCore-III CPU access
- *_PCIE registers apply to inbound PCIe access
- *_CSR registers apply to VCore-III SBA access block access

It is possible to enable weighted token arbitration scheme (WT_EN). When using this scheme, specific masters can be guaranteed a certain amount of bandwidth on the shared bus. Guaranteed bandwidth that is not used is given to other masters requesting the shared bus.

When weighted token arbitration is enabled, the masters on the shared bus are grated a configurable number of tokens (WT_CPU, WT_PCIE, and WT_CSR) at the start of each refresh period. The length of each refresh period is configurable (WT_TCL). For each clock cycle that the master uses the shared bus, the token counter for that master is decremented. When all tokens are spent, the master is forced to a low priority. Masters with tokens always take priority over masters with no tokens. The strict prioritized scheme is used to arbitrate between masters with tokens and between masters without tokens.

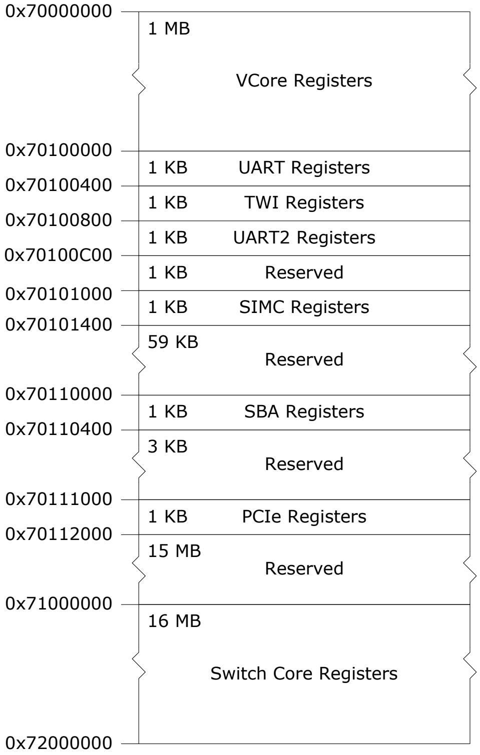
Example Guarantee That PCIe Can Get 25% Bandwidth. Configure WT_TCL to a refresh period of 2048 clock cycles; the optimal length of the refresh period depends on the scenario, experiment to find the right setting. Guarantee PCIe access in 25% of the refresh period by setting WT_PCIE to 512 (2048 × 25%). Set WT_CPU and WT_CSR to 0. This gives the VCore-III CPU and CSR unlimited tokens. Configure PCIe to the highest priority by setting PL_PCIe to 15. Finally, enable the weighted token scheme by setting WT_EN to 1. For each refresh period of 2048 clock cycles, PCIe is guaranteed access to the shared bus for 512 clock cycles because it is the highest priority master. When all the tokens are spent, it is put into the low-priority category.

3.3.2 CHIP REGISTER REGION

Registers in the VCore-III domain and inside the switch core are memory mapped into the chip registers region of the shared bus memory map. All registers are 32-bit wide and must only be accessed using 32-bit reads and writes. Bursts are supported.

Writes to this region are buffered (there is a one-word write buffer). Multiple back-to-back write access pauses the shared bus until the write buffer is freed up (until the previous writes are done). Reads from this region pause the shared bus until read data is available.

FIGURE 3-3: CHIP REGISTERS MEMORY MAP



The registers in the 0x70000000 through 0x70FFFFFF region are physically located inside the VCore-III system, so read and write access to these registers is fast (done in a few clock cycles). All registers in this region are considered fast registers.

Registers in the 0x71000000 through 0x71FFFFFF region are located inside the switch core; access to registers in this range takes approximately 1 µs. The DEVCPU_ORG and DEVCPU_QS targets are special; registers inside these two targets are faster; access to these two targets takes approximately 0.1 µs.

When more than one CPU is accessing registers, the access time may be increased. For more information, see Register Access and Multimaster Systems.

Writes to the Chip Registers region is buffered (there is a one-word write buffer). Multiple back-to-back write access pauses the shared bus until the write buffer is freed up (until the previous write is done). A read access pause the shared bus until read data is available. Executing a write immediately followed by a read requires the write to be done before the read can be started.

3.3.3 SI FLASH REGION

Read access from the SI Flash region initiates Flash formatted read access on the SI pins of the device by means of the SI boot controller.

The SI Flash region cannot be written to. Writing to the SI interface must be implemented by using the SI master controller. For more information, see Section 3.8.2, SI Master Controller. For legacy reasons, it is also possible to write to the SI interface by using the SI boot master’s software “bit-banging” register interface. For more information, see Section 3.8.1, SI Boot Controller.

3.3.4 DDR3/DDR3L REGION

Read and write access from or to the DDR3/DDR3L region maps to access on the DDR3 interface of the device using the DDR3 controller. For more information about the DDR3/DDR3L controller and initializing DDR3/DDR3L memory, see DDR3/DDR3L Memory Controller.

3.3.5 PCIE REGION

Read and write access from or to the PCIe region maps to outbound read and write access on the PCIe interface of the device by means of the PCIe endpoint controller. For more information about the PCIe endpoint controller and how to reach addresses in 32-bit and 64-bit PCIe environments, see PCIe Endpoint Controller.

3.4 VCore-III CPU

The VCore-III CPU system is based on a powerful MIPS24KEc-compatible microprocessor with 16-entry MMU, 32 kilobytes (kB) instruction, and 32 kB data caches.

This section describes how the VCore-III CPU is integrated into the VCore-III system. For more information about internal VCore-III CPU functions, such as bringing up caches, MMU, and so on, see the software board support package (BSP) at www.microsemi.com.

When the automatic boot in the little-endian or big-endian mode is enabled using the VCore-III strapping pins, the VCore-III CPU automatically starts to execute code from the SI Flash at byte address 0.

The following is a typical automatic boot sequence.

1. Speed up the boot interface. For more information, see SI boot controller.
2. Initialize the DDR3/DDR3L controller. For more information, see DDR3/DDR3L Memory Controller
3. Copy code-image from Flash to DDR3/DDR3L memory.
4. Change memory map from boot mode to normal mode, see Shared Bus.

When automatic boot is disabled, an external CPU is still able to start the VCore-III CPU through registers. For more information, see VCore Configurations.

The boot vector of the VCore-III CPU is mapped to the start of the KESEG1, which translates to physical address 0x00000000 on the VCore-III shared bus.

The VCore-III CPU interrupts are mapped to interrupt inputs 0 and 1, respectively.

3.4.1 LITTLE ENDIAN AND BIG ENDIAN SUPPORT

The VCore-III system is constructed as a little endian system, and registers descriptions reflect little endian encoding. When big endian mode is enabled, instructions and data are byte-lane swapped just before they enter and when they leave the VCore-III CPU. This is the standard way of translating between a CPU in big endian mode and a little endian system.

In big endian mode, care must be taken when accessing parts of the memory system that is also used by users other than the VCore-III CPU. For example, device registers are written and read by the VCore-III CPU, but they are also used by the device (which sees them in little endian mode). The VCore-III BSP contains examples of code that correctly handles register access for both little and big endian mode.

Endianess of the CPU is selected by VCore-III strapping pins or by register configuration when manually booting the CPU. For more information, see VCore-III configurations.

3.4.2 SOFTWARE DEBUG AND DEVELOPMENT

The VCore CPU has a standard MIPS EJTAG debug interface that can be used for breakpoints, loading of code, and examining memory. When the JTAG_ICE_nEN strapping pin is pulled low, the JTAG interface is attached to the EJTAG controller.

3.5 External CPU Support

An external CPU attaches to the device through the PCIe, SI, MIIM, or VRAP. Through these interfaces, an external CPU can access (and control) the device. For more information about interfaces and connections to device registers, see VCore-III system block diagram.

Inbound PCIe access is performed on the VCore Shared Bus (SBA) in the same way as ordinary VCore-III CPU access. By means of the switch core Access block it is possible to access the Switch Core Register (CSR) bus. For more information about supported PCIe BAR regions, see PCIe Endpoint Controller.

The SI, MIIM, and VRAP interfaces attach directly to the CSR. Through the VCore SBA access block, it is possible to access the VCore shared bus. For more information, see Access to the VCore Shared Bus.

The external CPU can coexist with the internal VCore-III CPU, and hardware-semaphores and interrupts are implemented for inter-CPU communication. For more information, see Section 3.5.5, Mailbox and Semaphores.

3.5.1 REGISTER ACCESS AND MULTIMASTER SYSTEMS

There are three different groups of registers in the device:

- Switch Core
- Fast Switch Core
- VCore

The Switch Core registers and Fast Switch Core registers are separated into individual register targets and attached to the Switch Core Register bus (CSR). The Fast Switch Core registers are placed in the DEVCPU_QS and DEVCPU_ORG register targets. Access to Fast Switch Core registers is less than 0.1 µs; other Switch Core registers take no more than 1 µs to access.

The VCore registers are attached directly to the VCore shared bus. The access time to VCore registers is negligible (a few clock cycles).

Although multiple masters can access VCore registers and Switch Core registers in parallel without noticeable penalty to the access time, the following exceptions apply.

- When accessing the same Switch Core register target (for example, DEVCPU_GCB), the second master to attempt access has to wait for the first master to finish (round robin arbitration applies.) This does not apply to Fast Switch Core register targets (DEVCPU_QS and DEVCPU_ORG).
- If both the VCore-IeVCore-III CPU and the PCIe master are performing Switch Core Register bus (CSR) access, both need to be routed through the Switch Core Access block. The second master has to wait for the first master to finish (shared bus arbitration applies).
- If two or more SI, MIIM, or VRAP masters are performing VCore register access, they all need to go through the VCore SBA Access block. Ownership has to be resolved by use of software (for example, by using the built-in semaphores).

The most common multimaster scenario is with an active VCore-III CPU and an external CPU using either SI or VRAP. In this case, Switch Core register access to targets that are used by both CPUs may see two times the access time (no more than 2 µs).

3.5.2 SERIAL INTERFACE IN SLAVE MODE

This section provides information about the function of the serial interface in slave mode.

The following table lists the registers associated with SI slave mode.

FIGURE 3-4: SI SLAVE MODE REGISTER

Register	Description
DEVCPU_ORG::IF_CTRL	Configuration of endianess and bit order
DEVCPU_ORG::IF_CFGSTAT	Configuration of padding
ICPU_CFG::GENERAL_CTRL	SI interface ownership

The serial interface implements an SPI-compatible protocol that allows an external CPU to perform read and write access to register targets within the device. Endianess and bit order is configurable, and several options for high frequencies are supported.

The serial interface is available to an external CPU when the VCore-III CPU does not use the SI for Flash or external SI access. For more information, VCore-III System and CPU interfaces.

The following table lists the serial interface pins when the SI slave is configured as owner of SI interface in GENERAL_CTRL.IF_SI_OWNER.

TABLE 3-3: SI SLAVE MODE PINS

Pin Name	I/O	Description
SI_nCS0	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 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, one don't care bit, 22 address bits, and 32 bits of write data (or don't care bits when reading).

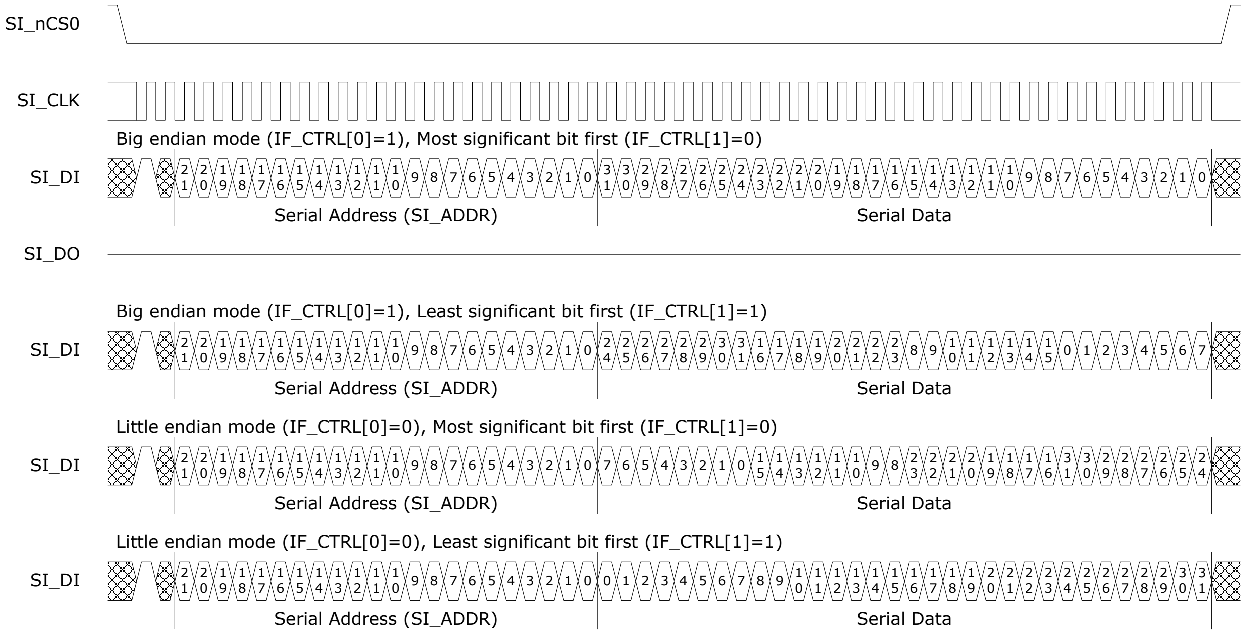
With the register address of a specific register (REG_ADDR), the SI address (SI_ADDR) is calculated as:

SI_ADDR = (REG_ADDR & 0x00FFFFFF)>>2

Data word endianess is configured through IF_CTRL[0]. The order of the data bits is configured using IF_CTRL[1].

The following illustration shows various configurations for 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 3-5: WRITE SEQUENCE FOR SI


SI_nCS0
SI_CLK
Big endian mode (IF_CTRL[0]=1), Most significant bit first (IF_CTRL[1]=0)
SI_DI
Serial Address (SI_ADDR)
Serial Data
SI_DO
Big endian mode (IF_CTRL[0]=1), Least significant bit first (IF_CTRL[1]=1)
SI_DI
Serial Address (SI_ADDR)
Serial Data
Little endian mode (IF_CTRL[0]=0), Most significant bit first (IF_CTRL[1]=0)
SI_DI
Serial Address (SI_ADDR)
Serial Data
Little endian mode (IF_CTRL[0]=0), Least significant bit first (IF_CTRL[1]=1)
SI_DI
Serial Address (SI_ADDR)
Serial Data

When using the serial interface to read registers, 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. For information about access time, see Register Access and Multimaster Systems. 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 master):
  1/(2 × 1 µs) = 500 kHz (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 6 MHz in a single master system. See the following 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 master), 1-byte padding give(1 × 8 – 1.5) /1 µs = 6 MHz (maximum). The SI_DO output is kept tri-stated until the actual read data is transmitted.

The following illustrations show options for serial read access. The illustrations show only one mapping of read data, 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 3-6: READ SEQUENCE FOR SI_CLK SLOW

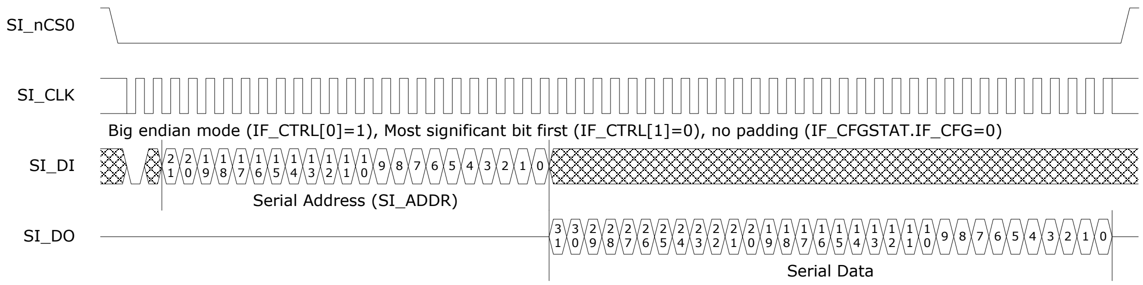

SI_nCS0
SI_CLK
Big endian mode (IF_CTRL[0]=1), Most significant bit first (IF_CTRL[1]=0), no padding (IF_CFGSTAT.IF_CFG=0)
SI_DI
Serial Address (SI_ADDR)
SI_DO
Serial Data

FIGURE 3-7: READ SEQUENCE FOR SI_CLK PAUSE

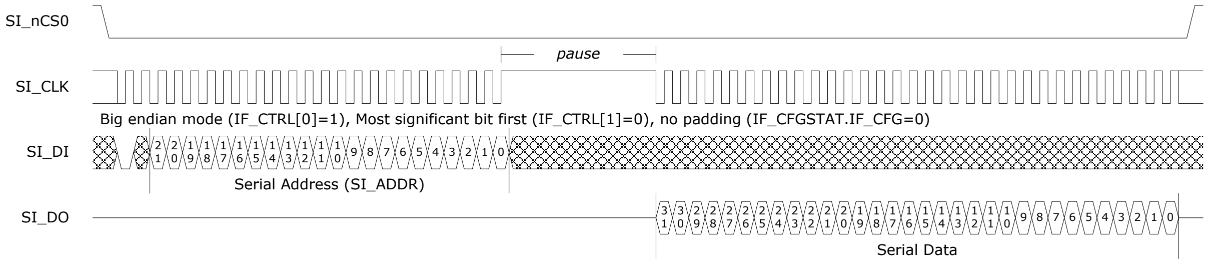
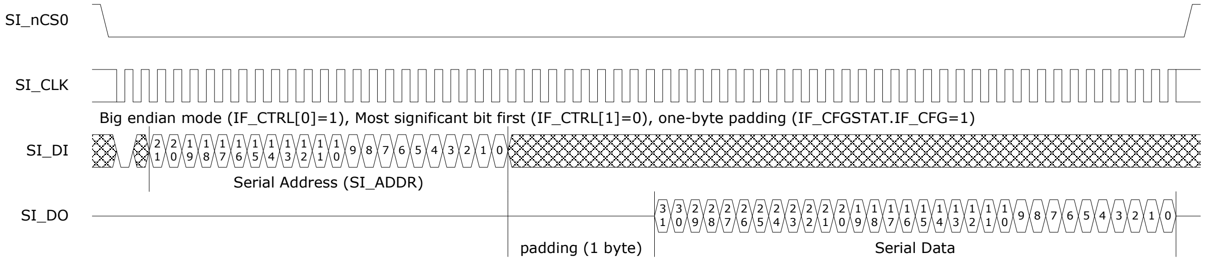

SI_nCS0
pause
SI_CLK
Big endian mode (IF_CTRL[0]=1), Most significant bit first (IF_CTRL[1]=0), no padding (IF_CFGSTAT.IF_CFG=0)
SI_DI
Serial Address (SI_ADDR)
SI_DO
Serial Data

FIGURE 3-8: READ SEQUENCE FOR ONE-BYTE PADDING


SI_nCS0
SI_CLK
Big endian mode (IF_CTRL[0]=1), Most significant bit first (IF_CTRL[1]=0), one-byte padding (IF_CFGSTAT.IF_CFG=1)
SI_DI
Serial Address (SI_ADDR)
SI_DO
padding (1 byte)
Serial Data

When dummy bytes are enabled (IF_CFGSTAT.IF_CFG), the SI slave logic enables an error check that sends out 0x88888888 and sets IF_CFGSTAT.IF_STAT if the SI master 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. For more information about constructing write data for this register, see the instructions in IF_CTRL.IF_CTRL.

3.5.3 MIIM INTERFACE IN SLAVE MODE

This section provides the functional aspects of the MIIM slave interface.

The MIIM slave interface allows an external CPU to perform read and write access to the device register targets. 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 slave pins on the device are overlaid functions on the GPIO interface. MIIM slave mode is enabled by configuring the appropriate VCore_CFG strapping pins. For more information, see Section 3.1, VCore-III Configurations. When MIIM slave mode is enabled, the appropriate GPIO pins are automatically overtaken. For more information about overlaid functions on the GPIOs for these signals, see Section 3.8.8.1, GPIO Overlaid Functions.

The following table lists the pins of the MIIM slave interface.

TABLE 3-4: MIIM SLAVE PINS

Pin Name	I/O	Description
MIIM_SLV_MDC/GPIO	I	MIIM slave clock input
MIIM_SLV_MDIO/GPIO	I/O	MIIM slave data input/output

MIIM_SLV_MDIO is sampled or changed on the rising edge of MIIM_SLV_MDC by the MIIM slave interface.

The MIIM slave can be configured to answer on one of two different PHY addresses using ICPU_CFG::GENERAL_CTRL.IF_MIIM_SLV_ADDR_SEL or the VCore_CFG strapping pins.

The MIIM slave has seven 16-bit MIIM registers defined as listed in the following table.

TABLE 3-5: MIIM 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.
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 slave using two MIIM write transactions. The two 16-bit data registers can then 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.

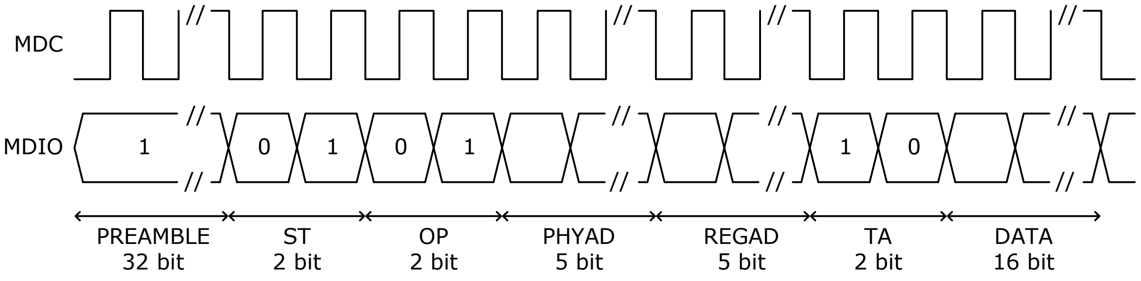
The address of the register to read/write is set in registers ADDR_REG0 and ADDR_REG1. The data to write to the register pointed to by the address in ADDR_REG0 and ADDR_REG1 is first written to DATA_REG0 and then to DATA_REG1. When the write transaction to DATA_REG1 is completed, the MIIM slave initiates 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 & 0x00FFFFFF)>>2

The following illustration shows a single MIIM write transaction on the MIIM interface.

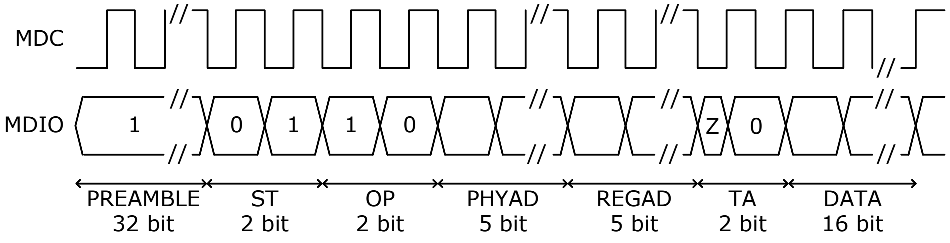
FIGURE 3-9: MIIM SLAVE WRITE SEQUENCE



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 before the DATA_REG1 register is read. In other words, the returned read value is from the previous read transaction.

The following illustration shows a single MIIM read transaction on the MIIM interface.

FIGURE 3-10: MIIM SLAVE READ SEQUENCE



3.5.4 ACCESS TO THE VCORE SHARED BUS

This section provides information about how to access the VCore shared bus (SBA) from an external CPU attached by means of the VRAP, SI, or MIIM. The following table lists the registers associated with the VCore shared bus access.

TABLE 3-6: VCORE SHARED BUS ACCESS REGISTERS

Register	Description
DEVCPU_GCB::VA_CTRL	Status for ongoing access
DEVCPU_GCB::VA_ADDR	Configuration of shared bus address
DEVCPU_GCB::VA_DATA	Configuration of shared bus address
DEVCPU_GCB::VA_DATA_INCR	Data register, access increments VA_ADDR
DEVCPU_GCB::VA_DATA_INERT	Data register, access does not start new access

An external CPU perform 32-bit reads and writes to the SBA through the VCore Access (VA) registers. In the VCore-III system, a dedicated master on the shared bus handles VA access. For information about arbitration between masters on the shared bus, see VCore Shared Bus Arbitration.

The SBA address is configured in VA_ADDR. Writing to VA_DATA starts an SBA write with the 32-bit value that was written to VA_DATA. Reading from VA_DATA returns the current value of the register and starts an SBA read access, when the read access completes, the result is automatically stored in the VA_DATA register.

The VA_DATA_INCR register behaves similar to VA_DATA, except that after starting an access, the VA_ADDR register is incremented by four (so that it points to the next word address in the SBA domain). Reading from the VA_DATA_INCR register returns the value of VA_DATA, writing to VA_DATA_INCR overwrites the value of VA_DATA.

Note By using VA_DATA_INCR, sequential word addresses can be accessed without having to manually increment the VA_ADDR register between each access.

The VA_DATA_INERT register provides direct access to the VA_DATA value without starting access on the SBA. Reading from the VA_DATA_INERT register returns the value of VA_DATA, writing to VA_DATA_INERT overwrites the value of VA_DATA.

The VCore-III shared bus is capable of returning error-indication when illegal register regions are accessed. If a VA access results in an error-indication from the SBA, the VA_CTRL.VA_ERR field is set, and the VA_DATA is set to 0x88888888.

Note SBA error indications only occur when non-existing memory regions or illegal registers are accessed. This will not happen during normal operation, so the VA_CTRL.VA_ERR indication is useful during debugging only.

Example Reading from ICPU_CFG::GPR[1] through the VA registers. The GPR register is the second register in the SBA VCore-III Registers region. Set VA_ADDR to 0x70000004, read once from VA_DATA (and discard the read value). Wait until VA_CTRL.VA_BUSY is cleared, then VA_DATA contains the value of the ICPU_CFG::GPR[1] register. Using VA_DATA_INERT (instead of VA_DATA) to read the data is appropriate, because this does not start a new SBA access.

3.5.4.1 Optimized Reading

VCore-III shared bus access is typically much faster than the CPU interface, which is used to access the VA registers. The VA_DATA register (VA_DATA_INCR and VA_DATA_INERT) return 0x88888888 while VA_CTRL.VA_BUSY is set. This means that it is possible to skip checking for busy between read access to SBA. For example, after initiating a read access from SBA, software can proceed directly to reading from VA_DATA, VA_DATA_INCR, or VA_DATA_INERT

- If the second read is different from 0x88888888, then the second read returned valid read data (the SBA access was done before the second read was performed).
- If the second read is equal to 0x88888888, then the VA logic may have been busy during the read and additional actions are required: First read the VA_CTRL.VA_BUSY field until the field is cleared (VA logic is not busy). Then read VA_DATA_INERT. If VA_DATA_INERT returns 0x88888888, then read-data is actually 0x88888888, continue to the next read. Otherwise, repeat the last read from VA_DATA, VA_DATA_INCR, or VA_DATA_INERT and then continue to the next read from there.

Optimized reading can be used for single read and sequential read access. For sequential reads, the VA_ADDR is only incremented on successful (non-busy) reads.

3.5.5 MAILBOX AND SEMAPHORES

This section provides information about the semaphores and mailbox features for CPU to CPU communication. The following table lists the registers associated with mailbox and semaphores.

TABLE 3-7: MAILBOX AND SEMAPHORE REGISTERS

Register	Description
DEVCPU_ORG::MAILBOX_SET	Atomic set of bits in the mailbox register.
DEVCPU_ORG::MAILBOX_CLR	Atomic clear of bits in the mailbox register.
DEVCPU_ORG::MAILBOX	Current mailbox state.
DEVCPU_ORG::SEMA_CFG	Configuration of semaphore interrupts.
DEVCPU_ORG::SEMA0	Taking of semaphore 0.
DEVCPU_ORG::SEMA1	Taking of semaphore 1.
DEVCPU_ORG::SEMA0_OWNER	Current owner of semaphore 0.
DEVCPU_ORG::SEMA1_OWNER	Current owner of semaphore 1.

The mailbox is a 32-bit register that can be set and cleared atomically using any CPU interface (including the VCore-III CPU). The MAILBOX register allows reading (and writing) of the current mailbox value. Atomic clear of individual bits in the mailbox register is done by writing a mask to MAILBOX_CLR. Atomic setting of individual bits in the mailbox register is done by writing a mask to MAILBOX_SET.

The device implements two independent semaphores. The semaphores are part of the Switch Core Register Bus (CSR) block and are accessible by means of the fast switch core registers. Semaphore ownership can be taken by interfaces attached to the CSR. That is, the VCore-III, VRAP, SI, and MIIM can be granted ownership. The VCore-IeVCore-III CPU and PCIe interface share the same semaphore, because they both access the CSR through the switch core access block.

Any CPU attached to an interface can attempt to take a semaphore n by reading SEMA0.SEMA0 or SEMA1.SEMA1. If the result is 1, the corresponding semaphore was successfully taken and is now owned by that interface. If the result is 0, the semaphore was not free. After successfully taking a semaphore, all additional reads from the corresponding register will return 0. To release a semaphore, write 1 to SEMA0.SEMA0 or SEMA1.SEMA1.

Note Any interface can release semaphores; it does not have to be the one that has taken the semaphore. This allows implementation of handshaking protocols.

The current status for a semaphore is available in SEMA0_OWNER.SEMA0_OWNER and SEMA1_OWNER.SEMA1_OWNER. See register description for encoding of owners.

Software interrupt is generated when semaphores are free or taken. Interrupt polarity is configured through SEMA_CFG.SEMA_INTR_POL. Semaphore 0 is hooked up to SW0 interrupt and semaphore 1 is hooked up to SW1 interrupt. For configuration of software-interrupt, see Interrupt Controller.

In addition to interrupting on semaphore state, software interrupt can be manually triggered by writing directly to the ICPU_CFG::INTR_FORCE register.

Software interrupts (SW0 and SW1) can be individually mapped to either the VCore-III CPU or to external interrupt outputs (to an external CPU).

3.6 PCIe Endpoint Controller

The device implements a single-lane PCIe 1.x endpoint that can be hooked up to any PCIe capable system. Using the PCIe interface, an external CPU can access (and control) the device. Ethernet frames can be injected and extracted using registers or the device can be configured to DMA Ethernet frames autonomously to/from PCIe memory. The device's DDR3/DDR3L memory is available through the PCIe interface, allowing the external CPU full read and write access to DDR3/DDR3L modules attached to the device. The DDR3/DDR3L controller and memory modules can be initialized either via PCIe or by the VCore-III CPU.

Both the VCore-III CPU and Frame DMA can generate PCIe read/write requests to any 32-bit or 64-bit memory region in PCIe memory space. However, it is up to software running on an external CPU to set up or communicate the appropriate PCIe memory mapping information to the device.

The defaults for the endpoints capabilities region and the extended capabilities region are listed in the registers list's description of the PCIE registers. The most important parameters are:

- Vendor (and subsystem vendor) ID: 0x101B, Microchip
- Device ID: 0xB005, device family ID
- Revision ID: 0x00, device family revision ID
- Class Code: 0x028000, Ethernet Network controller
- Single function, non-Bridge, INTA and Message Signaled Interrupt (MSI) capable device

The device family 0xB005 covers several register-compatible devices. The software driver must determine actual device ID and revision by reading DEVCPU_GCB::CHIP_ID from the device's memory mapped registers region.

For information about base address registers, see Base Address Registers, Inbound Requests.

The IDs, class code, revision ID, and base address register setups can be customized before enabling the PCIe endpoint. However, it requires a manual bring-up procedure by software running locally on the VCore-III CPU. For more information, see Enabling the Endpoint.

The endpoint is power management capable and implements PCI Bus Power Management Interface Specification Revision 1.2. For more information, see Power Management.

The endpoint is MSI capable. Up to four 64-bit messages are supported. Messages can be generated on rising and falling edges of each of the two external VCore-III interrupt destinations. For more information, see Outbound Interrupts.

For information about all PCI Express capabilities and extended capabilities register defaults, see the PCIE region's register descriptions.

3.6.1 ACCESSING ENDPOINT REGISTERS

The root complex accesses the PCIe endpoint's configuration registers by PCIe CfgRd/CfgWr requests. The VCore-III CPU can read configuration registers by means of the PCIE region. For more information, see Chip Register Region. The PCIE region must not be accessed when the PCIe endpoint is disabled.

The PCIe region is used during manual bring-up of PCIe endpoint. By using this region, it is possible to write most of the endpoint's read-only configuration registers, such as Vendor ID. The PCIe endpoint's read-only configuration register values must not be changed after the endpoint is enabled.

The VCore-III has a few dedicated PCIe registers in the ICPU_CFG:PCIE register group. An external CPU attached through the PCIe interface has to go through BAR0 to reach these registers.

3.6.2 ENABLING THE ENDPOINT

The PCIe endpoint is disabled by default. It can be enabled automatically or manually by either setting the VCore_CFG strapping pins or by software running in the VCore-III CPU.

The recommended approach is using VCore_CFG strapping pins, because it is fast and does not require special software running on the VCore-III CPU. The endpoint is ready approximately 50 ms after release of the device's nRESET. Until this point, the device ignores any attempts to do link training, and the PCIe output remains idle (tri-stated).

By using software running on the VCore-III CPU, it is possible to manually start the PCIe endpoint. Note that PCIe standard specifies a maximum delay from nRESET release to working PCIe endpoint so software must enable the endpoint as part of the boot process.

The root complex must follow standard PCIe procedures for bringing up the endpoint.

3.6.2.1 Manually Starting MAC/PCS and SerDes

This section provides information about how to manually start up the PCIe endpoint and customize selected configuration space parameters.

The following table lists the registers related to manually bringing up PCIe.

TABLE 3-8: MANUAL PCIE BRING-UP REGISTERS

Register	Description
ICPU_CFG::PCIE_CFG	Disable of automatic link initialization
HSIO::SERDES6G_COMMON_CFG	SERDES configuration
HSIO::SERDES6G_MISC_CFG	SERDES configuration
HSIO::SERDES6G_IB_CFG	SERDES configuration
HSIO::SERDES6G_IB_CFG1	SERDES configuration
HSIO::SERDES6G_OB_CFG	SERDES configuration
HSIO::SERDES6G_OB_CFG1	SERDES configuration
HSIO::SERDES6G_DES_CFG	SERDES configuration
HSIO::SERDES6G_PLL_CFG	SERDES configuration
HSIO::SERDES6G_MISC_CFG	SERDES configuration
HSIO::MCB_SERDES6G_ADDR_CFG	SERDES configuration
PCIE::DEVICE_ID_VENDOR_ID, PCIE::CLASS_CODE_REVISION_ID, PCIE::SUBSYSTEM_ID_SUBSYSTEM_VENDOR_ID	Device parameter customization
PCIE::BAR1, PCIE::BAR2	Base address register customization

Disable automatic link training for PICe endpoint.

1. PCIE_CFG.LTSSM_DIS = 1.

Configure and enable PCIe SERDES for 2G5 mode.

1. SERDES6G_MISC_CFG = 0x00000031.
2. SERDES6G_OB_CFG = 0x60000171.
3. SERDES6G_OB_CFG1 = 0x000000B0.
4. SERDES6G_DES_CFG = 0x000068A6.
5. SERDES6G_IB_CFG = 0x3D57AC37.
6. SERDES6G_IB_CFG1 = 0x00110FF0.
7. SERDES6G_COMMON_CFG = 0x00024009.
8. MCB_SERDES6G_ADDR_CFG = 0x80000004 and wait until bit 31 is cleared.
9. SERDES6G_PLL_CFG = 0x00030F20.
10. MCB_SERDES6G_ADDR_CFG = 0x80000004 and wait until bit 31 is cleared.
11. Wait at least 20 ms.
12. SERDES6G_IB_CFG = 0x3D57AC3F.
13. SERDES6G_MISC_CFG = 0x00000030.

14. MCB_SERDES6G_ADDR_CFG = 0x80000004 and wait until bit 31 is cleared.
15. Wait at least 60 ms.
16. SERDES6G_IB_CFG = 0x3D57ACBF.
17. SERDES6G_IB_CFG1 = 0x00103FF0.
18. MCB_SERDES6G_ADDR_CFG = 0x80000004 and wait until bit 31 is cleared.

To optionally disable BAR1:

1. PCIE_CFG.PCIE_BAR_WR_ENA = 1
2. BAR1 = 0x00000000
3. PCIE_CFG.PCIE_BAR_WR_ENA = 0

To optionally increase BAR2 from 128 to 256 megabytes:

1. PCIE_CFG.PCIE_BAR_WR_ENA = 1
2. BAR2 = 0x0FFFFFFF
3. PCIE_CFG.PCIE_BAR_WR_ENA = 0

To optionally change selected PCIe configuration space values:

- Write Vendor ID/Device ID using DEVICE_ID_VENDOR_ID.
- Write Class Code/Revision ID using CLASS_CODE_REVISION_ID.
- Write Subsystem ID/Subsystem Vendor ID using SUBSYSTEM_ID_SUBSYSTEM_VENDOR_ID.

Enable automatic link training for PCIe endpoint

1. PCIE_CFG.LTSSM_DIS = 0

The last step enables the endpoint for link training, and the root complex will then be able to initialize the PCIe endpoint. After this the PCIE parameters must not be changed anymore.

3.6.3 BASE ADDRESS REGISTERS INBOUND REQUESTS

The device implements three memory regions. Read and write operations using the PCIe are translated directly to read and write access on the SBA. When manually bringing up the PCIe endpoint, BAR1 can be disabled. For more information, see Manually Starting MAC/PCS and SerDes.

TABLE 3-9: BASE ADDRESS REGISTERS

Register	Description
BAR0, 32-bit, 32 megabytes	Chip registers region. This region maps to the Chip registers region in the SBA address space. See Chip Register Region. This region only supports 32-bit word-aligned reads and writes. Single and burst accesses are supported.
BAR1, 32-bit, 16 megabytes	SI Flash region. This region maps to the SI Flash region in the SBA address space. See SI Flash Region. This region only supports 32-bit word-aligned reads. Single and burst access is supported.
BAR2, 32-bit, 128 megabytes	DDR3/DDR3L region. This region maps to the low 128 megabytes of the DDR3/DDR3L region in the SBA address space. See DDR3/DDR3L Region.

This region supports all access types.

To access the BAR1 region, a Flash must be attached to the SI interface of the device. For information about how to set up I/O timing and to program the Flash through BAR0; the Chip Registers Region, see Section 3.8.1, SI Boot Controller.

To access the BAR2 region, DDR3/DDR3L modules must be present and initialized. For information about initializing the DDR3/DDR3L controller and modules through BAR0; the Chip Registers, see DDR3/DDR3L Memory Controller. During manual bring up of PCIe, the size of the BAR2 region can be changed to match the size of the attached DDR3/DDR3L modules. For more information, see Manually Starting MAC/PCS and SerDes.

3.6.4 OUTBOUND INTERRUPTS

The device supports both Message Signaled Interrupt (MSI) and Legacy PCI Interrupt Delivery. The root complex configures the desired mode using the MSI enable bit in the PCIe MSI Capability Register Set. For information about the VCore-III interrupt controller, see Section 3.8.13, Interrupt Controller.

The following table lists the device registers associated PCIe outbound interrupts.

TABLE 3-10: PCIE OUTBOUND INTERRUPT REGISTERS

Register	Description
ICPU_CFG::PCIE_INTR_COMMON_CFG	Interrupt mode and enable
ICPU_CFG::PCIE_INTR_CFG	MSI parameters

In legacy mode, one interrupt is supported; select either EXT_DST0 or EXT_DST1 using PCIE_INTR_COMMON_CFG.LEGASY_MODE_INTR_SEL. The PCIe endpoint uses Assert_INTA and Deassert_INTA messages when configured for legacy mode.

In MSI mode, both EXT_DST interrupts can be used. EXT_DST0 is configured though PCIE_INTR_CFG[0] and EXT_DST1 through PCIE_INTR_CFG[1]. Enable message generation on rising and/or falling edges in PCIE_INTR_CFG[*n*].INTR_RISING_ENA and PCIE_INTR_CFG[*n*].INTR_FALLING_ENA. Different vectors can be generated for rising and falling edges, configure these through PCIE_INTR_CFG[*n*].RISING_VECTOR_VAL and PCIE_INTR_CFG[*n*].FALLING_VECTOR_VAL. Finally, each EXT_DST interrupt must be given an appropriate traffic class through PCIE_INTR_CFG[*n*].TRAFFIC_CLASS.

After the root complex has configured the PCIe endpoint's MSI Capability Register Set and the external CPU has configured how interrupts from the VCore-III interrupt controller are propagated to PCIe, interrupts must then be enabled by setting PCIE_INTR_COMMON_CFG.PCIE_INTR_ENA.

3.6.5 OUTBOUND ACCESS

After the PCIe endpoint is initialized, outbound read/write access to PCIe memory space is initiated by reading or writing the SBA's PCIe DMA region.

The following table lists the device registers associated with PCIe outbound access.

TABLE 3-11: OUTBOUND ACCESS REGISTERS

Register	Description
ICPU_CFG::PCIESLV_SBA	Configures SBA outbound requests.
ICPU_CFG::PCIESLV_FDMA	Configures FDMA outbound requests.
PCIE::ATU_REGION	Select active region for the ATU_* registers.
PCIE::ATU_CFG1	Configures TLP fields.
PCIE::ATU_CFG2	Enable address translation.
PCIE::ATU_TGT_ADDR_LOW	Configures outbound PCIe address.
PCIE::ATU_TGT_ADDR_HIGH	Configures outbound PCIe address.

The PCIe DMA region is 1 gigabyte. Access in this region is mapped to any 1 gigabyte region in 32-bit or 64-bit PCIe memory space by using address translation. Two address translation regions are supported. The recommended approach is to configure the first region for SBA outbound access and the second region for FDMA outbound access.

Address translation works by taking bits [29:0] from the SBA/FDMA address, adding a configurable offset, and then using the resulting address to access into PCIe memory space. Offsets are configurable in steps of 64 kilobytes in the ATU_TGT_ADDR_HIGH and ATU_TGT_ADDR_LOW registers.

The software on the VCore-III CPU (or other SBA masters) can dynamically reconfigure the window as needed; however, the FDMA does not have that ability, so it must be disabled while updating the 1 gigabyte window that is set up for it.

Note Although the SBA and FDMA both access the PCIe DMA region by addresses 0xC0000000 though 0xFFFFFFFF, the PCIe address translation unit can differentiate between these accesses and apply the appropriate translation.

3.6.5.1 Configuring Outbound SBA Translation Region

Configure PCIESLV_SBA.SBA_OFFSET = 0 and select address translation region 0 by writing ATU_REGION.ATU_IDX = 0. Set PCIE::ATU_BASE_ADDR_LOW.ATU_BASE_ADDR_LOW = 0x0000 and PCIE::ATU_LIMIT_ADDR.ATU_LIMIT_ADDR = 0x3FFF.

The following table lists the appropriate PCIe headers that must be configured before using the SBA's PCIe DMA region. Remaining header fields are automatically handled.

TABLE 3-12: PCIE ACCESS HEADER FIELDS

Header Field	Register::Fields
Attributes	ATU_CFG1.ATU_ATTR
Poisoned Data	PCIESLV_SBA.SBA_EP
TLP Digest Field Present	ATU_CFG1.ATU_TD
Traffic Class	ATU_CFG1.ATU_TC
Type	ATU_CFG1.ATU_TYPE
Byte Enables	PCIESLV_SBA.SBA_BE
Message Code	ATU_CFG2.ATU_MSG_CODE

Configure the low address of the destination window in PCIe memory space as follows:

- Set ATU_TGT_ADDR_HIGH.ATU_TGT_ADDR_HIGH to bits [63:32] of the destination window. Set to 0 when a 32-bit address must be generated.
- Set ATU_TGT_ADDR_LOW.ATU_TGT_ADDR_LOW to bits [31:16] of the destination window. This field must not be set higher than 0xC000.

Enable address translation by writing ATU_CFG2.ATU_REGION_ENA = 1. SBA access in the PCIe DMA region is then mapped to PCIe memory space as defined by ATU_TGT_ADDR_LOW and ATU_TGT_ADDR_HIGH.

The header fields and the PCIe address fields can be reconfigured on-the-fly as needed; however, set ATU_REGION.ATU_IDX to 0 to ensure that the SBA region is selected.

3.6.5.2 Configuring Outbound FDMA Translation Region

Configure PCIESLV_FDMA.FDMA_OFFSET = 1, and select address translation region 1 by writing ATU_REGION.ATU_IDX = 1.

Set PCIE::ATU_BASE_ADDR_LOW.ATU_BASE_ADDR_LOW = 0x4000 and PCIE::ATU_LIMIT_ADDR.ATU_LIMIT_ADDR = 0x7FFF.

The FDMA PCIe header must be MRd/MWr type, reorderable, cache-coherent, and without ECRC. Remaining header fields are automatically handled.

TABLE 3-13: FDMA PCIE ACCESS HEADER FIELDS

Header Field	Register::Fields	Suggested Value
Attributes	ATU_CFG1.ATU_ATTR	Set to 2
TLP Digest Field Present	ATU_CFG1.ATU_TD	Set to 0
Traffic Class	ATU_CFG1.ATU_TC	Use an appropriate traffic class
Type	ATU_CFG1.ATU_TYPE	Set to 0

Configure low address of destination window in PCIe memory space as follows:

- Set ATU_TGT_ADDR_HIGH.ATU_TGT_ADDR_HIGH to bits [63:32] of the destination window. Set to 0 when a 32-bit address must be generated.
- Set ATU_TGT_ADDR_LOW.ATU_TGT_ADDR_LOW to bits [31:16] of the destination window. This field must not be set higher than 0xC000.

Enable address translation by writing ATU_CFG2.ATU_REGION_ENA = 1. The FDMA can be configured to make access in the 0xC0000000 - 0xFFFFFFFF address region. These accesses will then be mapped to PCIe memory space as defined by ATU_TGT_ADDR_LOW and ATU_TGT_ADDR_HIGH. The FDMA must be disabled if the address window needs to be updated.

3.6.6 POWER MANAGEMENT

The device's PCIe endpoint supports D0, D1, and D3 device power-management states and associated link power-management states. The switch core does not automatically react to changes in the PCIe endpoint's power management states. It is, however, possible to enable a VCore-III interrupt on device power state changes and then have the VCore-III CPU software make application-specific changes to the device operation depending on the power management state.

TABLE 3-14: POWER MANAGEMENT REGISTERS

Register	Description
ICPU_CFG::PCIE_STAT	Current power management state
ICPU_CFG::PCIEPCS_CFG	Configuration of WAKE output and beacon
ICPU_CFG::PCIE_INTR	PCIe interrupt sticky events
ICPU_CFG::PCIE_INTR_ENA	Enable of PCIe interrupts
ICPU_CFG::PCIE_INTR_IDENT	Currently interrupting PCIe sources
ICPU_CFG::PCIE_AUX_CFG	Configuration of auxiliary power detection

Because the device does not implement a dedicated auxiliary power for the PCIe endpoint, the endpoint is operated from the VDD core power supply. Before the power management driver initializes the device, software can "force" auxiliary power detection by writing PCIE_AUX_CFG = 3, which causes the endpoint to report that it is capable of emitting Power Management Events (PME) messages in the D3c state.

The current device power management state is available using PCIE_STAT.PM_STATE. A change in this field's value sets the PCIE_INTR.INTR_PM_STATE sticky bit. To enable this interrupt, set PCIE_INTR_ENA.INTR_PM_STATE_ENA. The current state of the PCIe endpoint interrupt towards the VCore-III interrupt controller is shown in PCIE_INTR_IDENT register (if different from zero, then interrupt is active).

The endpoint can emit PMEs if the PME_En bit is set in the PM Capability Register Set and if the endpoint is in powerdown mode.

- Outbound request from either SBA or FDMA trigger PME.
- A change in status for an enabled outbound interrupt (either legacy or MSI) triggers PME. This feature can be disabled by setting ICPU_CFG::PCIE_INTR_COMMON_CFG.WAKEUP_ON_INTR_DIS.

In the D3 state, the endpoint transmits a beacon. The beacon function can be disabled and instead drive the WAKE output using the overlaid GPIO function. For more information about the overlaid function on the GPIO for this signal, see Section 3.8.8.1, GPIO Overlaid Functions.

TABLE 3-15: PCIE WAKE PIN

Pin Name	I/O	Description
PCIe_WAKE/GPIO	O	PCIe WAKE output

Enable WAKE by setting PCIEPCS_CFG.BEACON_DIS. The polarity of the WAKE output is configured in PCIEPCS_CFG.WAKE_POL. The drive scheme is configured in PCIEPCS_CFG.WAKE_OE.

3.6.7 DEVICE RESET USING PCIE

The built-in PCIe reset mechanism in the PCIe endpoint resets only the PCIe MAC. The device reset is not tied into the MAC reset. To reset the complete the device, use the following procedure.

1. Save the state of the PCIe controller registers using operating system.
2. Set DEVCPU_GCB::SOFT_RST.SOFT_CHIP_RST.
3. Wait for 100 ms.
4. Recover state of PCIe controller registers using operating system.

Setting SOFT_CHIP_RST will cause re-initialization of the device.

3.7 Frame DMA

This section describes the Frame DMA engine (FDMA). When FDMA is enabled, Ethernet frames can be extracted or injected autonomously to or from the device's DDR3/DDR3L memory and/or PCIe memory space. Linked list data structures in memory are used for injecting or extracting Ethernet frames. The FDMA generates interrupts when frame extraction or injection is done and when the linked lists needs updating.

TABLE 3-16: FDMA REGISTERS

Register	Description
DEVCPU_QS::XTR_GRP_CFG	CPU port ownership, extraction direction.
DEVCPU_QS::INJ_GRP_CFG	CPU port ownership, injection direction.
DEVCPU_QS::INJ_CTRL	Injection EOF to SOF spacing.
SYS::PORT_MODE	Enables IFH and disables FCS recalculation (for injected frames).
ICPU_CFG::FDMA_CH_CFG	Channel configuration, priorities, and so on.
ICPU_CFG::FDMA_CH_ACTIVATE	Enables channels.
ICPU_CFG::FDMA_CH_DISABLE	Disables channels.
ICPU_CFG::FDMA_CH_STAT	Status for channels.
ICPU_CFG::FDMA_CH_SAFE	Sets when safe to update channel linked lists.
ICPU_CFG::FDMA_DCB_LLP	Linked list pointer for channels.
ICPU_CFG::FDMA_DCB_LLP_PREV	Previous linked list pointer for channels.
ICPU_CFG::FDMA_CH_CNT	Software counters for channels.
ICPU_CFG::FDMA_INTR_LLP	NULL pointer event for channels.
ICPU_CFG::FDMA_INTR_LLP_ENA	Enables interrupt on NULL pointer event.
ICPU_CFG::FDMA_INTR_FRM	Frame done event for channels.
ICPU_CFG::FDMA_INTR_FRM_ENA	Enables interrupt on frame done event.
ICPU_CFG::FDMA_INTR_SIG	SIG counter incremented event for channels.
ICPU_CFG::FDMA_INTR_SIG_ENA	Enables interrupt on SIG counter event.
ICPU_CFG::MANUAL_INTR	Manual injection/extraction events.
ICPU_CFG::MANUAL_INTR_ENA	Enables interrupts on manual injection/extraction events.
ICPU_CFG::FDMA_EVT_ERR	Error event for channels.
ICPU_CFG::FDMA_EVT_ERR_CODE	Error event description.
ICPU_CFG::FDMA_INTR_ENA	Enables interrupt for channels.
ICPU_CFG::FDMA_INTR_IDENT	Currently interrupting channels.
DEVCPU_QS::XTR_FRM_PRUNING	Enables pruning of extraction frames.
ICPU_CFG::FDMA_CH_INJ_TOKEN_CNT	Injection tokens.
ICPU_CFG::FDMA_CH_INJ_TOKEN_TICK_RLD	Periodic addition of injection tokens.
ICPU_CFG::MANUAL_CFG	Configures manual injection/extraction.
ICPU_CFG::MANUAL_XTR	Memory region used for manual extraction.
ICPU_CFG::MANUAL_INJ	Memory region used for manual injection.
ICPU_CFG::FDMA_GCFG	Configures injection buffer watermark.

The FDMA implements two extraction channels per CPU port and a total of eight injection channels. Extraction channels are hard-coded per CPU port, and injection channels can be individually assigned to any CPU port.

- FDMA channel 0 corresponds to port 11(group 0) extraction direction.
- FDMA channel 1 corresponds to port 12(group 1) extraction direction.
- FDMA channel 2 through 9 corresponds to port 11 (group 0) injection direction when FDMA_CH_CFG[channel].CH_INJ_GRP is set to 0.
- FDMA channel 2 through 9 corresponds to port 12 (group 1) injection direction when FDMA_CH_CFG[chan-

nel].CH_INJ_GRP is set to 1.

The FDMA implements a strict priority scheme. Injection and extraction channels can be assigned individual priorities, which are used when the FDMA has to decide between servicing two or more channels. Channel priority is configured in FDMA_CH_CFG[ch].CH_PRIO. When channels have same priority, the higher channel number takes priority over the lower channel number.

When more than one injection channel is enabled for injection on the same CPU port, then priority determines which channel that is allowed to inject data. Ownership is re-arbitrated on frame boundaries.

The internal frame header is added in front of extracted frames and provides useful switching information about the extracted frames.

Injection frames requires an internal frame header for controlling injection parameters. The internal frame header is added in front of frame data. The device recalculates and overwrites the Ethernet FCS for frames that are injected through the CPU when requested by setting in the internal frame header.

For more information about the extraction and injection IFH, see Section 2.14, CPU Port Module.

The FDMA supports a manual mode where the FDMA decision logic is disabled and the FDMA takes and provides data in the order that was requested by an external master (internal or external CPU). The manual mode is a special case of normal FDMA operation where only few of the FDMA features apply. For more information about manual operation, see Manual Mode.

The following configuration must be performed before enabling FDMA extraction: Set XTR_GRP_CFG[group].MODE = 2.

The following configurations must be performed before enabling FDMA injection:

Set INJ_GRP_CFG[group].MODE = 2. Set INJ_CTRL[group].GAP_SIZE = 0,

set PORT_MODE[port].INCL_INJ_HDR = 1.

3.7.1 DMA CONTROL BLOCK STRUCTURES

The FDMA processes linked lists of DMA Control Block Structures (DCBs). The DCBs have the same basic structure for both injection or for extraction. A DCB must be placed on a 32-bit word-aligned address in memory. Each DCB must have an associated data block that is placed on a 32-bit word aligned address in memory, the length of the data block must be a complete number of 32-bit words.

An Ethernet frame can be contained inside one data block (if the data block is big enough) or the frame can be spread across multiple data blocks. A data block never contains more than one Ethernet frame. Data blocks that contain start-of-frame have set a special bit in the DCB's status word, likewise for data blocks that contains end of frame. The FDMA stores or retrieves Ethernet frame data in network order. This means that the data at byte address (*n*) of a frame was received just before the data at byte address (*n* + 1).

Frame data inside the DCB's associated data blocks can be placed at any byte offset and have any byte length as long as byte offset and length does not exceed the size of the data block. Byte offset and length is configured using special fields inside the DCB status word. Software can specify offset both when setting up extraction and injection DCBs. Software only specifies length for injection DCBs; the FDMA automatically calculates and updates length for extraction.

Example If a DCB's status word has block offset 5 and block length 2, then the DCB's data block contains two bytes of frame data placed at second and third bytes inside the second 32-bit word of the DCB's associated data block.

DCBs are linked together by the DCB's LLP field. The last DCB in a chain must have LLP = NULL. Chains consisting of a single DCB are allowed.

FIGURE 3-11: FDMA DCB LAYOUT

<table>
<tr><td>word-address *n*</td><td colspan="7">LLP [31:0]: Linked List Pointer, this field points to the next DCB in the list, set to NULL for end of list. DCBs must be word-aligned, which means LLP must point to a word-aligned address (bit [1:0] = 00).</td><td>LLP</td></tr>
<tr><td>*n* + 1</td><td colspan="7">DATAP [31:0]: Data block pointer, this field points to the data block associated with this DCB. DATAP must be !=NULL and must point to a word-aligned address (bit [1:0] = 00).</td><td>DATAP</td></tr>
<tr><td>*n* + 2</td><td>[31:24]: For SW use. FDMA will not modify.</td><td colspan="3">[23:18]: Must be 0.</td><td>17</td><td>16</td><td>DATAL [15:0]: Number of bytes available in the data block associated with this DCB. DATAL must be a complete number of words (bit [1:0] = 00).</td><td>DATAL</td></tr>
<tr><td>*n* + 3</td><td colspan="2">BLOCKO [31:20]: Byte offset from beginning of the data block to the first byte of frame-data.</td><td>19</td><td>18</td><td>17</td><td>16</td><td>BLOCKL [15:0]: Number of frame-data bytes in the data block. BLOCKL does not include offset bytes.</td><td>STAT</td></tr>
</table>

STAT[16] SOF: Set to 1 if data block contains start of frame.

STAT[17] EOF: Set to 1 if data block contains end of frame.

STAT[18] ABORT: Abort indication.
Extraction: Set to 1 if the frame associated with the data block was aborted.
Injection: If set to 1 when FDMA loads the DCB, it aborts the frame associated with the data block.

STAT[19] PD: Pruned/Done indication.
Extraction: Set to 1 if the frame associated with the data block was pruned.
Injection: The FDMA set this to 1 when done processing the DCB. If set to 1 when FDMA loads the DCB, it is treated as ABORT.

DATAL[16] SIG: If set to 1 when FDMA loads the DCB, the CH_CNT_SIG counter is incremented by one.

DATAL[17] TOKEN: Token indication, only used during injection.
If set to 1, the FDMA uses one token (CH_INJ_TOKEN_CNT) when injecting the contents of the DCB. If the token counter is 0 when loading the DCB, then injection is postponed until tokens are made available.

3.7.2 ENABLING AND DISABLING FDMA CHANNELS

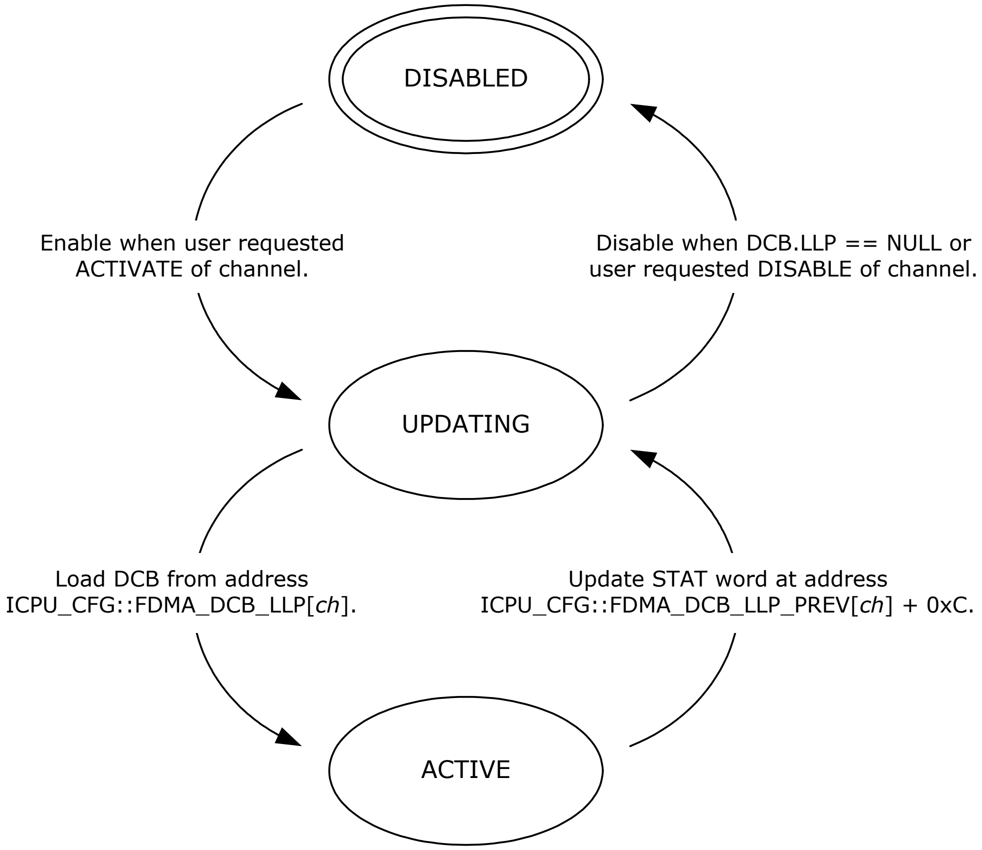
To enable a channel (ch), write a valid DCB pointer to FDMA_DCB_LLP[ch] and enable the channel by setting FDMA_CH_ACTIVATE.CH_ACTIVATE[ch]. This makes the FDMA load the DCB from memory and start either injection or extraction.

To schedule a channel for disabling, set FDMA_CH_DISABLE.CH_DISABLE[ch]. An active channel does not disable immediately. Instead, it waits until the current data block is done, saves the status word, and then disables.

Channels can be in one of three states: DISABLED, UPDATING, or ACTIVE. Channels are DISABLED by default. When the channel is reading a DCB or writing the DCB status word, it is considered to be UPDATING. The status of individual channels is available in FDMA_CH_STAT.CH_STAT[ch].

The following illustration shows the FDMA channel states.

FIGURE 3-12: FDMA CHANNEL STATES


DISABLED
Enable when user requested ACTIVATE of channel.
Disable when DCB.LLP == NULL or user requested DISABLE of channel.
UPDATING
Load DCB from address ICPU_CFG::FDMA_DCB_LLP[*ch*].
Update STAT word at address ICPU_CFG::FDMA_DCB_LLP_PREV[*ch*] + 0xC.
ACTIVE

A channel that has FDMA_DCB_LLP[ch].DCB_LLP==NULL when going from ACTIVE to UPDATING disables itself instead of loading a new DCB. After this it can be re-enabled as previously described. Extraction channels emit an INTR_LLP-event when loading a DCB with LLP==NULL. Injection channels emit an INTR_LLP-event when saving status for a DCB that has the LLP==NULL.

Note: Extraction channel running out of DCBs during extraction is a problem that software must avoid. A hanging extraction channel will potentially be head-of-line blocking other extraction channels.

It is possible to update an active channels LLP pointer and pointers in the DCB chains. Before changing pointers software must schedule the channel for disabling (by writing FDMA_CH_DISABLE.CH_DISABLE[ch]) and then wait for the channel to set FDMA_CH_SAFE.CH_SAFE[ch]. When the pointer update is complete, soft must re-activate the channel by setting FDMA_CH_ACTIVATE.CH_ACTIVATE[ch]. Setting activate will cancel the deactivate-request, or if the channel has disabled itself in the meantime, it will re activate the channel.

Note: The address of the current DCB is available in FDMA_DCB_LLP_PREV[ch]. This information is useful when modifying pointers for an active channel. The FDMA does not reload the current DCB when re-activated, so if the LLP-field of the current DCB is modified, then software must also modify FDMA_DCB_LLP[ch].

Setting FDMA_CH_CFG[ch].DONEEOF_STOP_ENA disables an FDMA channel and emits LLP-event after saving status for DCBs that contains EOF (after extracting or injecting a complete frame). Setting FDMA_CH_CFG[ch].DONE_STOP_ENA disables an FDMA channel and emits LLP-event after saving status for any DCB.

3.7.3 CHANNEL COUNTERS

The FDMA implements three counters per channel: SIG, DCB, and FRM. These counters are accessible through FDMA_CH_CNT[ch].CH_CNT_SIG, FDMA_CH_CNT[ch].CH_CNT_DCB, and FDMA_CH_CNT[ch].CH_CNT_FRM, respectively. For more information about how to safely modify these counters, see the register descriptions.

- The SIG (signal) counter is incremented by one each time the FDMA loads a DCB that has the DATAL.SIG bit set to 1.
- The FRM (frame) counter is incremented by one each time the FDMA store status word for DCB that has EOF set.

It is a wrapping counter that can be used for software driver debug and development. This counter does not count aborted frames.

- The DCB counter is incremented by one every time the FDMA loads a DCB from memory. It is a wrapping counter that can be used for software driver debug and development.

It is possible to enable channel interrupt whenever the SIG counter is incremented; this makes it possible for software to receive interrupt when the FDMA reaches certain points in a DCB chain.

3.7.4 FDMA EVENTS AND INTERRUPTS

Each FDMA channel can generate four events: LLP-event, FRM-event, SIG-event, and ERR-event. These events cause a bit to be set in FDMA_INTR_LLP.INTR_LLP[ch], FDMA_INTR_FRM.INTR_FRM[ch], FDMA_INTR_SIG.INTR_SIG[ch], and FDMA_EVT_ERR.EVT_ERR[ch], respectively.

- LLP-event occurs when an extraction channel loads a DCB that has LLP = NULL or when an injection channel writes status for a DCB that has LLP=NULL. LLP-events are also emitted from channels that have FDMA_CH_CFG[ch].DONEEOF_STOP_ENA or FDMA_CH_CFG[ch].DONE_STOP_ENA set. For more information, see Enabling and Disabling FDMA Channels.
- FRM-event is indicated when an active channel loads a new DCB and the previous DCB had EOF. The FRM-event is also indicated for channels that are disabled after writing DCB status with EOF.
- SIG-event is indicated whenever the FDMA_CH_CNT[ch].CH_CNT_SIG counter is incremented. The SIG (signal) counter is incremented when loading a DCB that has the DATAL.SIG bit set.
- ERR-event is an error indication that is set for a channel if it encounters an unexpected problem during normal operation. This indication is implemented to ease software driver debugging and development. A channel that encounters an error will be disabled, depending on the type of error the channel state may be too corrupt for it to be restarted without system reset. When an ERR-event occurs, the FDMA_EVT_ERR_CODE.EVT_ERR_CODE shows the exact reason for an ERR-event. For more information about the errors that are detected, see FDMA_EVT_ERR_CODE.EVT_ERR_CODE.

Each of the events (LLP, FRM, SIG) can be enabled for channel interrupt through FDMA_INTR_LLP_ENA.INTR_LLP_ENA[ch], FDMA_INTR_FRM_ENA.INTR_FRM_ENA[ch], and FDMA_INTR_SIG.INTR_SIG[ch] respectively. The ERR event is always enabled for channel interrupt.

The highest numbered extraction channel supports two additional non-sticky events related to manual extraction: XTR_SOF_RDY-event and XTR_ANY_RDY-event. An active event causes the following fields to be set: MANUAL_INTR.INTR_XTR_SOF_RDY and MANUAL_INTR.INTR_XTR_ANY_RDY respectively. For more information, see Manual Extraction.

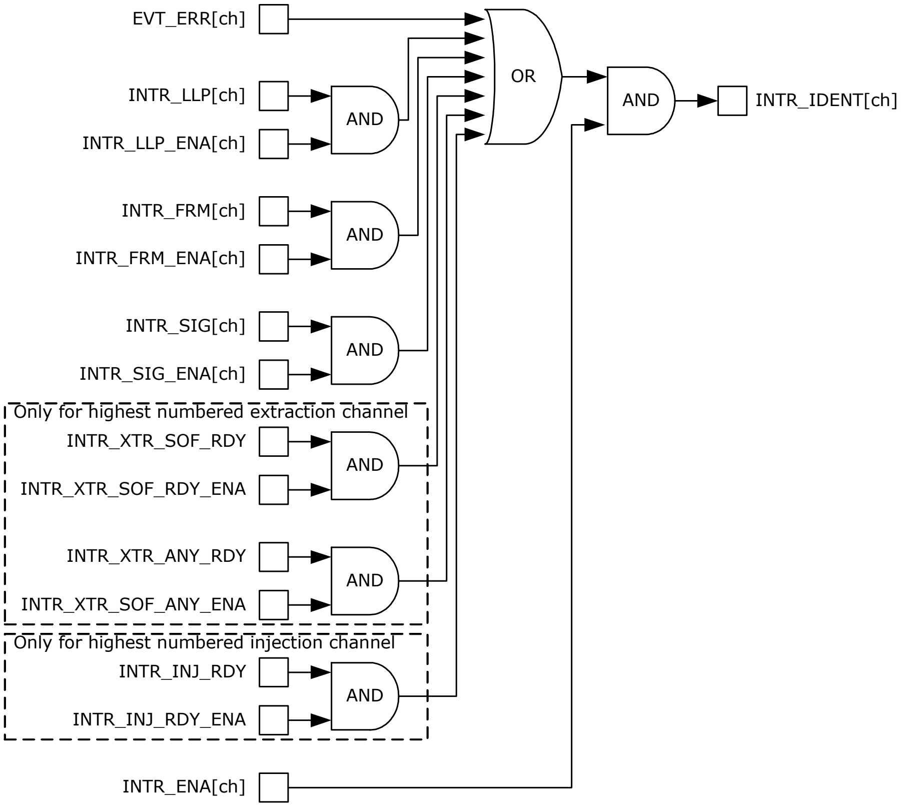
- XTR_SOF_RDY-event is active when the next word to be manually extracted contains an SOF indication. This event is enabled in MANUAL_INTR_ENA.INTR_XTR_SOF_RDY_ENA.
- XTR_ANY_RDY-event is active when any word is available for manually extraction. This event is enabled in MANUAL_INTR_ENA.INTR_XTR_ANY_RDY_ENA respectively.

The highest numbered injection channel supports one additional non-sticky event related to manual injection: INJ_RDY-event. This event is active when the injection logic has room for (at least) sixteen 32-bit words of injection frame data. When this event is active, MANUAL_INTR.INTR_INJ_RDY is set. The INJ_RDY-event can be enabled for channel interrupt by setting MANUAL_INTR_ENA.INTR_INJ_RDY_ENA. For more information, see Manual Injection.

The FDMA_INTR_ENA.INTR_ENA[ch] field enables interrupt from individual channels, FDMA_INTR_IDENT.INTR_IDENT[ch] field shows which channels that are currently interrupting. While INTR_IDENT is non-zero, the FDMA is indicating interrupting towards to the VCore-III interrupt controller.

The following illustration shows the FDMA channel interrupt hierarchy.

FIGURE 3-13: FDMA CHANNEL INTERRUPT HIERARCHY



3.7.5 FDMA EXTRACTION

During extraction, the FDMA extracts Ethernet frame data from the Queuing System and saves it into the data block of the DCB that is currently loaded by the FDMA extraction channel. The FDMA continually processes DCBs until it reaches a DCB with LLP = NULL or until it is disabled.

When an extraction channel writes the status word of a DCB, it updates SOF/EOF/ABORT/PRUNED-indications and BLOCKL. BLOCKO remains unchanged (write value is taken from the value that was read when the DCB was loaded).

Aborting frames from the queuing system will not occur during normal operation. If the queuing system is reset during extraction of a particular frame, then ABORT and EOF is set. Software must discard frames that have ABORT set.

Pruning of frames during extraction is enabled per extraction port through XTR_FRM_PRUNING[group].PRUNE_SIZE. When enabled, Ethernet frames above the configured size are truncated, and PD and EOF is set.

3.7.6 FDMA INJECTION

During injection, the FDMA reads Ethernet frame data from the data block of the DCB that is currently loaded by the FDMA injection channel and injects this into the queuing system. The FDMA continually processes DCBs until it reaches a DCB with LLP = NULL or until it is disabled.

When an injection channel writes the status word of a DCB, it sets DONE indication. All other status word fields remain unchanged (write value is stored the first time the injection channel loads the DCB).

The rate by which the FDMA injects frames can be shaped by using tokens. Each injection channel has an associated token counter (FDMA_CH_INJ_TOKEN_CNT[ich]). A DCB that has the DATAL.TOKEN field set causes the injection channel to deduct one from the token counter before the data block of the DCB can be injected. If the token counter is at 0, the injection channel postpones injection until the channels token counter is set to a value different from 0.

Tokens can be added to the token counter by writing the FDMA_CH_INJ_TOKEN_CNT[ich] register. Tokens can also be added automatically (with a fixed interval) by using the dedicated token tick counter. Setting FDMA_CH_INJ_TOKEN_TICK_RLD[ich] to a value *n* (different from 0) will cause one token to be added to that injection channel every *n* x 200 ns.

3.7.7 MANUAL MODE

The decision making logic of the FDMA extraction path and/or injection paths can be disabled to give control of the FDMA's extraction and/or injection buffers directly to any master attached to the Shared Bus Architecture. When operating in manual mode DCB structure, counters and most of the interrupts do not apply.

Manual extraction and injection use hard-coded channel numbers.

- Manual extraction mode uses FDMA channel 1 (port 12 extraction direction).
- Manual injection mode uses FDMA channel 9 (port 12 injection direction).

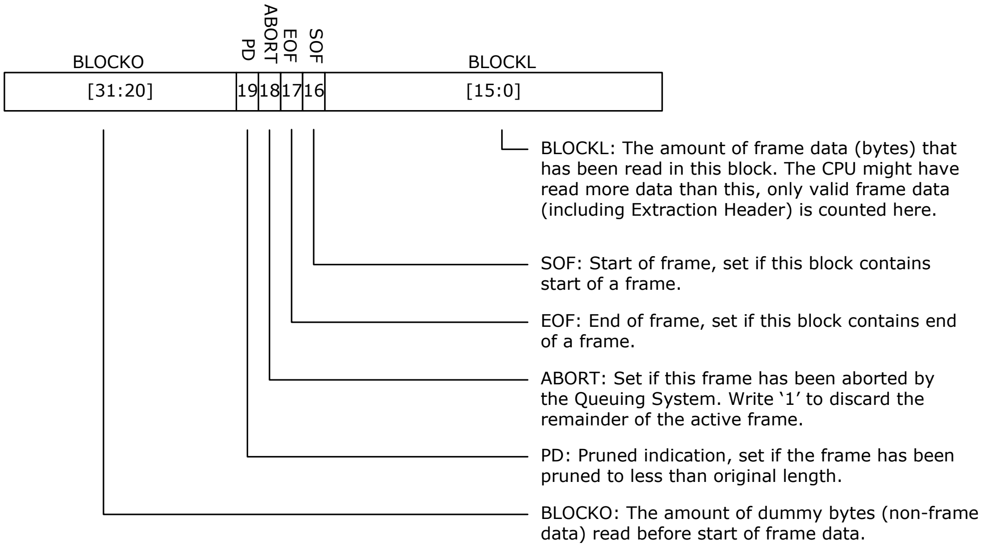
To enable manual extraction, set MANUAL_CFG.XTR_ENA. The FDMA must not be enabled for FDMA extraction when manual extraction mode is active. To enable manual injection, set MANUAL_CFG.INJ_ENA and FDMA_CH_CFG[9].CH_INJ_CRP = 1. The FDMA must not be enabled for FDMA injection when manual injection mode is active. Extraction and injection paths are independent. For example, FDMA can control extraction at the same time as doing manual injection by means of the CPU.

3.7.7.1 Manual Extraction

Extraction is done by reading one or more blocks of data from the extraction memory area. A block of data is defined by reading one or more data words followed by reading of the extraction status word. The extraction memory area is 16 kilobytes and is implemented as a replicated register region MANUAL_XTR. The highest register in the replication (0xFFF) maps to the extraction status word. The status word is updated by the extraction logic and shows the number of frame bytes read, SOF and EOF indications, and PRUNED/ABORT indications.

Note During frame extraction, the CPU does not know the frame length. This means that the CPU must check for EOF regularly during frame extraction. When reading a data block from the device, the CPU can burst read from the memory area in such a way that the extraction status word is read as the last word of the burst.

FIGURE 3-14: EXTRACTION STATUS WORD ENCODING


BLOCKO
PD
ABORT
EOF
SOF
BLOCKL
[31:20]
19
18
17
16
[15:0]
BLOCKL: The amount of frame data (bytes) that has been read in this block. The CPU might have read more data than this, only valid frame data (including Extraction Header) is counted here.
SOF: Start of frame, set if this block contains start of a frame.
EOF: End of frame, set if this block contains end of a frame.
ABORT: Set if this frame has been aborted by the Queuing System. Write '1' to discard the remainder of the active frame.
PD: Pruned indication, set if the frame has been pruned to less than original length.
BLOCKO: The amount of dummy bytes (non-frame data) read before start of frame data.

The extraction logic updates the extraction status word while the CPU is reading out data for a block. Prior to starting a new data block, the CPU can write the two least significant bits of the BLOCKO field. The BLOCKO value is stored in the extraction logic and takes effect when the new data block is started. Reading the status field always returns the BLOCKO value that applies to the current data block. Unless written (before stating a data block), the BLOCKO is cleared, so by default all blocks have 0 byte offset. The offset can be written multiple times; the last value takes effect.

The CPU can abort frames that it has already started to read out by writing the extraction status field with the ABORT field set. All other status-word fields will be ignored. This causes the extraction logic to discard the active frame and remove it from the queuing system.

Reading of block data does not have to be done in one burst. As long as the status word is not read, multiple reads from the extraction region are treated as belonging to the same block.

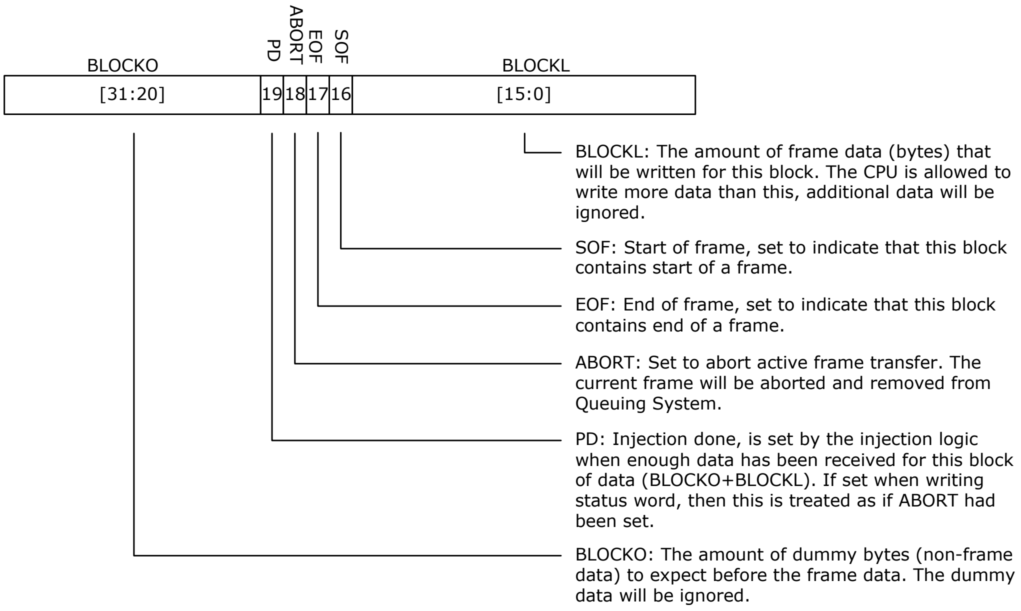
3.7.7.2 Manual Injection

Injection is done by writing one or more blocks of data to the injection memory area. A block of data is defined by writing an injection status word followed by writing one or more data words. The injection memory area is 16 kilobytes and is implemented as a replicated register region MANUAL_INJ. The first register in this replication maps to the injection status word. The status word for each block defines the following:

- Number of bytes to be injected
- Optional byte offset before injection data
- SOF and EOF indications
- Optional ABORT indication

Note In general, it makes sense to inject frames as a single large block of data (containing both SOF and EOF). However, because offset/length can be specified individually for each block, injecting frames through several blocks is useful when compensating for offset/word misalignment. For example, when building a frame from different memory regions in the CPU main memory.

FIGURE 3-15: INJECTION STATUS WORD ENCODING


BLOCKO
PD
ABORT
EOF
SOF
BLOCKL
[31:20]
19
18
17
16
[15:0]
BLOCKL: The amount of frame data (bytes) that will be written for this block. The CPU is allowed to write more data than this, additional data will be ignored.
SOF: Start of frame, set to indicate that this block contains start of a frame.
EOF: End of frame, set to indicate that this block contains end of a frame.
ABORT: Set to abort active frame transfer. The current frame will be aborted and removed from Queuing System.
PD: Injection done, is set by the injection logic when enough data has been received for this block of data (BLOCKO+BLOCKL). If set when writing status word, then this is treated as if ABORT had been set.
BLOCKO: The amount of dummy bytes (non-frame data) to expect before the frame data. The dummy data will be ignored.

Injection logic updates the PD field of the status word. The status word can be read at any time during injection without affecting current data block transfers. However, a CPU is able to calculate how much data that is required to inject a complete block of data (at least BLOCKO + BLOCKL bytes), so reading the status word is for software development and debug only. Writing status word before finishing a previous data block will cause that frame to be aborted. Frames are also aborted if they do not start with SOF or end with EOF indications.

As long as the status word is not written, multiple writes to the injection region are treated as data belonging to the same block. This means multiple bursts of data words can be written between each injection status word update.

Software can abort frames by writing to injection status with ABORT field set. All other status word fields are ignored. When a frame is aborted, the already injected data is removed from the queuing system.

3.8 VCore-III System Peripherals

This section describes the subblocks of the VCore-III system. Although the subblocks are primarily intended to be used by the VCore-III CPU, an external CPU can also access and control them through the shared bus.

3.8.1 SI BOOT CONTROLLER

The SPI boot master allows booting from a Flash that is connected to the serial interface. For information about how to write to the serial interface, see Section 3.8.2, SI Master Controller. For information about using an external CPU to access device registers using the serial interface, see Section 3.5.2, Serial Interface in Slave Mode.

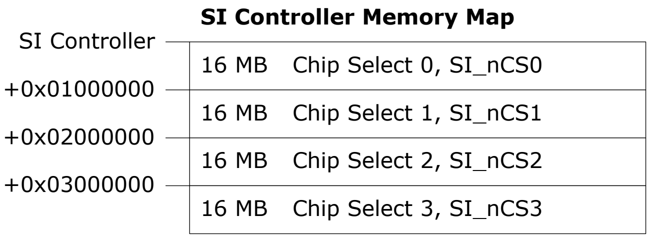
The following table lists the registers associated with the SI boot controller.

TABLE 3-17: SI BOOT CONTROLLER CONFIGURATION REGISTERS

Register	Description
ICPU_CFG::SPI_MST_CFG	Serial interface speed and address width
ICPU_CFG::SW_MODE	Legacy manual control of the serial interface pins
ICPU_CFG::GENERAL_CTRL	SI interface ownership

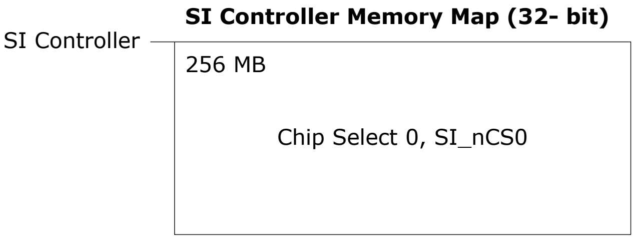
By default, the SI boot controller operates in 24-bit address mode. In this mode, there are four programmable chip selects when the VCore-III system controls the SI. Each chip select can address up to 16 megabytes (MB) of memory.

FIGURE 3-16: SI BOOT CONTROLLER MEMORY MAP IN 24-BIT MODE


SI Controller Memory Map
SI Controller
+0x01000000
+0x02000000
+0x03000000
16 MB Chip Select 0, SI_nCS0
16 MB Chip Select 1, SI_nCS1
16 MB Chip Select 2, SI_nCS2
16 MB Chip Select 3, SI_nCS3

The SI boot controller can be reconfigured for 32-bit address mode through SPI_MST_CFG.A32B_ENA. In 32-bit mode, the entire SI region of 256 megabytes (MB) is addressed using chip select 0.

FIGURE 3-17: SI BOOT CONTROLLER MEMORY MAP IN 32-BIT MODE



Reading from the memory region for a specific SI chip select generates an SI read on that chip select. The VCore-III CPU can execute code directly from Flash by executing from the SI boot controller's memory region. For 32-bit capable SI Flash devices, the VCore-III must start up in 24-bit mode. During the boot process, it must manually reconfigure the Flash device (and SI boot controller) into 32-bit mode and then continue booting.

The SI boot controller accepts 8-bit, 16-bit, and 32-bit read access with or without bursting. Writing to the SI requires manual control of the SI pins using software. Setting SW_MODE.SW_PIN_CTRL_MODE places all SI pins under software control. Output enable and the value of SI_CLK, SI_DO, SI_nCS*n* are controlled through the SW_MODE register. The value of the SI_DI pin is available in SW_MODE.SW_SPI_SDI. The software control mode is provided for legacy reasons; new implementations should use the dedicated master controller for writing to the serial interface. For more information see Section 3.8.2, SI Master Controller.

Note The VCore-III CPU cannot execute code directly from the SI boot controller's memory region at the same time as manually writing to the serial interface.

The following table lists the serial interface pins when the SI boot controller is configured as owner of SI interface in GENERAL_CTRL.IF_SI_OWNER. For more information about overlaid functions on the GPIOs for these signals, see Section 3.8.8.1, GPIO Overlaid Functions.

TABLE 3-18: SERIAL INTERFACE PINS

Pin Name	I/O	Description
SI_nCS0 SI_nCS[3:1]/GPIO	O	Active low chip selects. Only one chip select can be active at any time. Chip selects 1 through 3 are overlaid functions on GPIO pins.

TABLE 3-18: SERIAL INTERFACE PINS (CONTINUED)

Pin Name	I/O	Description
SI_CLK	O	Clock output.
SI_DO	O	Data output (MOSI).
SI_DI	I	Data input (MISO).

The SI boot controller does speculative pre-fetching of data. After reading address *n*, the SI boot controller automatically continues reading address *n* + 1, so that the next value is ready if requested by the VCore-III CPU. This greatly optimizes reading from sequential addresses in the Flash, such as when copying data from Flash into program memory.

The following illustrations depict 24-bit address mode. When the controller is set to 32-bit mode (through SPI_MST_CFG.A32B_ENA), 32 address bits are transferred instead of 24.

FIGURE 3-18: SI READ TIMING IN NORMAL MODE

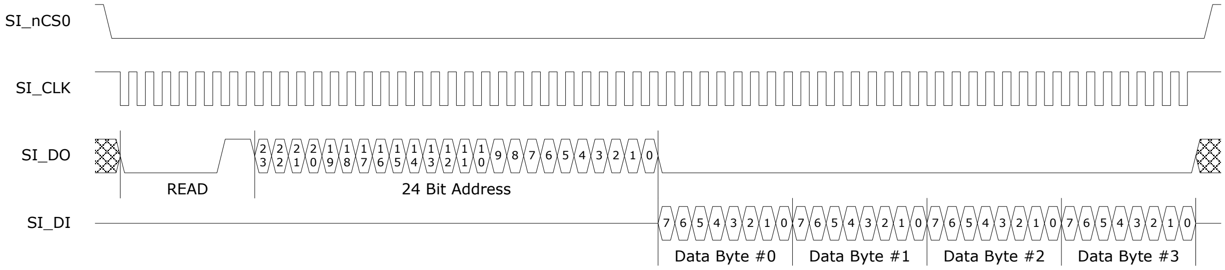
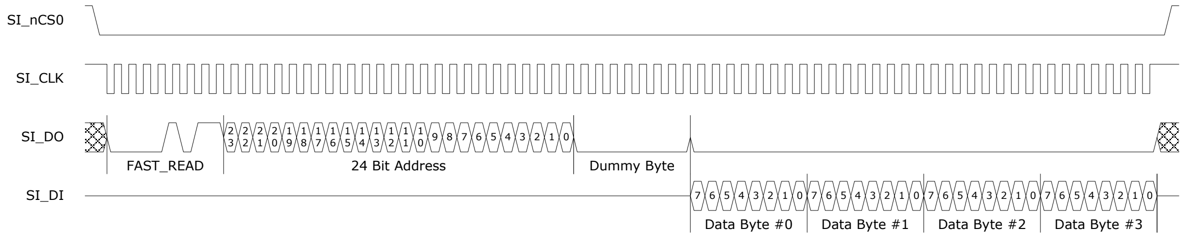


FIGURE 3-19: SI READ TIMING IN FAST MODE



The default timing of the SI boot controller operates with most serial interface Flash devices. Use the following process to calculate the optimized serial interface parameters for a specific SI device:

1. Calculate an appropriate frequency divider value as described in SPI_MST_CFG.CLK_DIV. The SI operates at no more than 25 MHz, and the maximum frequency of the SPI device must not be exceeded. The VCore-III system frequency in the device is 250 MHz.
2. The SPI device may require a FAST_READ command rather than normal READ when the SI frequency is increased. Setting SPI_MST_CFG.FAST_READ_ENA makes the SI boot controller use FAST_READ commands.
3. Calculate SPI_MST_CFG.CS_DESELECT_TIME so that it matches how long the SPI device requires chip-select to be deasserted between access. This value depends on the SI clock period that results from the SPI_MST_CFG.CLK_DIV setting.

These parameters must be written to SPI_MST_CFG. The CLK_DIV field must either be written last or at the same time as the other parameters. The SPI_MST_CFG register can be configured while also booting up from the SI.

When the VCore-III CPU boots from the SI interface, the default values of the SPI_MST_CFG register are used until the SI_MST_CFG is reconfigured with optimized parameters. This implies that SI_CLK is operating at approximately 8.1 MHz, with normal read instructions and maximum gap between chip select operations to the Flash.

Note The SPI boot master does optimized reading. SI_DI (from the Flash) is sampled just before driving falling edge on SI_CLK (to the Flash). This greatly relaxes the round trip delay requirement for SI_CLK to SI_DI, allowing high Flash clock frequencies.

3.8.2 SI MASTER CONTROLLER

This section describes the SPI master controller (SIMC) and how to use it for accessing external SPI slave devices, such as programming of a serially attached Flash device on the boot interface. VCore booting from serial Flash is handled by the SI boot master.

The following table lists the registers associated with the SI master controller.

TABLE 3-19: SI MASTER CONTROLLER CONFIGURATION REGISTERS OVERVIEW

Register	Description
SIMC::CTRLR0	Transaction configuration
SIMC::CTRLR1	Configurations for receive-only mode
SIMC::SIMCEN	SI master controller enable
SIMC::SER	Slave select configuration
SIMC::BAUDR	Baud rate configuration
SIMC::TXFTLR	Tx FIFO threshold level
SIMC::RXFTLR	Rx FIFO Threshold Level
SIMC::TXFLR	Tx FIFO fill level
SIMC::RXFLR	Rx FIFO fill level
SIMC::SR	Various status indications
SIMC::IMR	Interrupt enable
SIMC::ISR	Interrupt sources
SIMC::RISR	Unmasked interrupt sources
SIMC::TXOICR	Clear of transmit FIFO overflow interrupt
SIMC::RXOICR	Clear of receive FIFO overflow interrupt
SIMC::RXUICR	Clear of receive FIFO underflow interrupt
SIMC::DR	Tx/Rx FIFO access
SIMC::RX_SAMPLE_DLY	RXD sample delay
ICPU_CFG::GENERAL_CTRL	Interface configurations

The SI master controller supports Motorola SPI and Texas Instruments SSP protocols. The default protocol is SPI, enable SSP by setting CTRLR0.FRF = 1 and GENERAL_CTRL.SSP_MODE_ENA = 1. The protocol baud rate is programmable in BAUDR.SCKDV; the maximum baud rate is 25 MHz.

Before the SI master controller can be used, it must be set as owner of the serial interface. This is done by writing GENERAL_CTRL.IF_SI_OWNER = 2.

The SI master controller has a programmable frame size. The frame size is the smallest unit when transferring data over the SI interface. Using CTRLR0.DFS, the frame size is configured in the range of 4 bits to 16 bits. When reading or writing words from the transmit/receive FIFO, the number of bits that is stored per FIFO-word is always equal to frame size (as programmed in CTRLR0.DFS).

The controller operates in one of three following major modes: Transmit and receive, transmit only, or receive only. The mode is configured in CTRLR0.TMOD.

Transmit and receive. software paces SI transactions. For every data frame that software writes to the transmission FIFO, another data frame is made available in the receive FIFO (receive data from the serial interface). Transaction will go on for as long as there is data available in the transmission FIFO.

Transmit only. Software paces SI transactions. The controller transmits only; receive data is discarded. Transaction will go on for as long as there is data available in the transmission FIFO.

Receive only. The controller paces SI transactions. The controller receives only data, software requests a specific number of data frames by means of CTRLR1.NDF, the transaction will go on until all data frames has been read from the SI interface. Transaction is initiated by software writing one data frame to transmission FIFO. This frame is not used for anything else than starting the transaction. The SI_DO output is undefined during receive only transfers. Receive data frames are put into the receive FIFO as they are read from SI.

For SPI mode, chip select is asserted throughout transfers. Transmit transfers end when the transmit FIFO runs out of data frames. Software must make sure it is not interrupted while writing data for a multi-frame transfer. When multiple distinct transfers are needed, wait for SR.TFE = 1 and SR.BUSY = 0 before initiating new transfer. SR.BUSY is an indication of ongoing transfers, however, it is not asserted immediately when writing frames to the FIFO. As a result, software must also check the SR.TFE (transmit FIFO empty) indication.

The SI master controller supports up to 5 chip selects. Chip select 0 is mapped to the SI_nCS pin of the serial interface. The remaining chips selects are available using overlaid GPIO functions. Software controls which chip selects to activate for a transfer using the SER register. For more information about overlaid functions on the GPIOs for these signals, see Section 3.8.8.1, GPIO Overlaid Functions.

TABLE 3-20: SI MASTER CONTROLLER PINS

Pin Name	I/O	Description
SI_nCS0 SI_nCS[4:1]/GPIO	O	Active low chip selects. Chip selects 1 through 4 are overlaid functions on the GPIO pins.
SI_CLK	O	Clock output.
SI_DO	O	Data output (MOSI).
SI_DI	I	Data input (MISO).

Writing to any DR replication index put data into the transmission FIFO, and reading from any DR replication index take out data from the receive FIFO. FIFO status indications are available in the SR register’s TFE, TFNF, RFF, and RFNE fields. For information about interrupting on FIFO status, see Section 3.8.2.4, SIMC Interrupts.

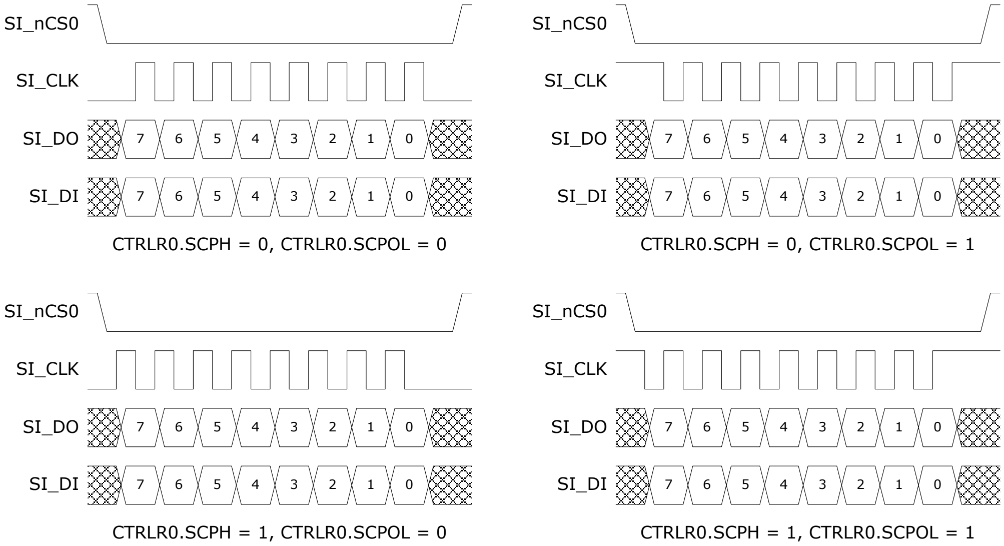
It is important to delay the sample of the RXD input signal by writing to SIMC::RX_SAMPLE_DLY. Each value represents a single VCore system clock cycle delay on the sample. If this field is programmed with a value that exceeds 25, a zero (0) delay will be applied. For optimal operation, it is suggested to set the value to SIMC::BAUD/2, but no larger than 25. This will delay sampling of RX-data by half clock cycle for SPI frequencies above 5MHz. Below 5 MHz the sampling delay will be 100ns. It is impossible to write to this register when the SIMC is enabled.

After completing all initial configurations, the SI master controller is enabled by writing SIMCEN.SIMCEN = 1. Some registers can only be written when the controller is in disabled state. This is noted in the register-descriptions for the affected registers.

3.8.2.1 SPI Protocol

When the SI master controller is configured for SPI mode, clock polarity and position is configurable in CTRLR0.SCPH and CTRLR0.SCPOL. The following illustration shows the four possible combinations for 8-bit transfers.

FIGURE 3-20: SIMC SPI CLOCK CONFIGURATIONS

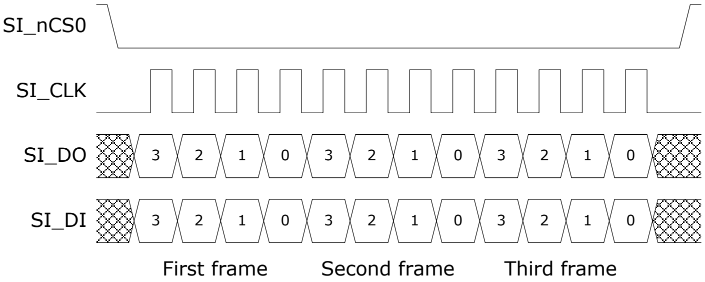

SI_nCS0
SI_CLK
SI_DO
SI_DI
7 6 5 4 3 2 1 0
CTRLR0.SCPH = 0, CTRLR0.SCPOL = 0
SI_nCS0
SI_CLK
SI_DO
SI_DI
CTRLR0.SCPH = 0, CTRLR0.SCPOL = 1
SI_nCS0
SI_CLK
SI_DO
SI_DI
CTRLR0.SCPH = 1, CTRLR0.SCPOL = 0
SI_nCS0
SI_CLK
SI_DO
SI_DI
CTRLR0.SCPH = 1, CTRLR0.SCPOL = 1

Data is sampled in the middle of the data-eye.

When using SPI protocol and transferring more than one data frame at the time, the controller performs one consecutive transfer. The following illustration shows a transfer of three data frames of 4 bits each.

Note Transmitting transfers end when the transmit FIFO runs out of data frames. Receive only transfers end when the pre-defined number of data-frames is read from the SI interface.

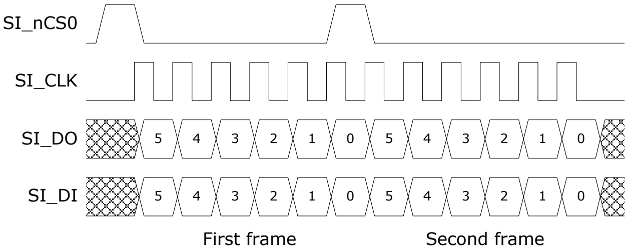
FIGURE 3-21: SIMC SPI 3X TRANSFERS



3.8.2.2 SSP Protocol

The SSP protocol for transferring two 6-bit data frames is shown in the following illustration. When using SSP mode, CTRLR0.SCPH and CTRLR0.SCPOL must remain zero.

3.8.2.3 SIMC SSP 2x Transfers



3.8.2.4 SIMC Interrupts

The SI master controller has five interrupt sources:

- RXF interrupt is asserted when the fill level of the receive FIFO (available in RXFLR) exceeds the programmable level set in RXFTLR. This interrupt is non-sticky; it is deasserted when fill level is decreased below the programmable level.
- RXO interrupt is asserted on receive FIFO overflow. This interrupt is cleared by reading the RXOICR register.
- RXU interrupt is asserted when reading from an empty receive FIFO. This interrupt is cleared by reading the RXUICR register.
- TXO interrupt is asserted when writing to a full transmit FIFO. This interrupt is cleared by reading the TXOICR register.
- TXE interrupt is asserted when the fill level of the transmit FIFO (available in TXFLR) is below the programmable level set in TXFTLR. This interrupt is non-sticky; it is deasserted when fill level is increased above the programmable level.

The raw (unmasked) status of these interrupts is available in the RISR register. Each of the interrupts can be individually masked by the IMR register. All interrupts are enabled by default. The currently active interrupts are shown in the ISR register.

If the RXO, RXU, and TXO interrupts occur during normal use of the SI master controller, it is an indication of software problems.

Interrupt is asserted towards the VCore-III interrupt controller when any enabled interrupt is asserted and the SI master controller is enabled.

3.8.2.5 SIMC Programming Example

This example shows how to use the SI master controller to interface with the SI slave of another device. The slave device will be initialized for big endian/most significant bit first mode and then the DEVCPU_GCB::GPR register is written with the value 0x01234567. It is assumed that the other device’s SI slave is connected on SI_nCS1 (and that appropriate GPIO alternate function has been enabled).

The SI slave is using a 24-bit address field and a 32-bit data field. This example uses 4 x 16-bit data frames to transfer the write-access. This means that 8 bits too much will be transferred, this is not a problem for the SI slave interface; it ignores data above and beyond the 56 bit required for a write-access.

For information about initialization and how to calculate the 24-bit SI address field, see Section 3.5.2, Serial Interface in Slave Mode. The address-field when writing DEVCPU_GCB::GPR is 0x804001 (including write-indication).

The following steps are required to bring up the SI master controller and write twice to the SI slave device.

Important The following procedure disconnects the SI boot master from the SI interface. Booting must be done before attempting to overtake the boot-interface.

1. Write GENERAL_CTRL.IF_SI_OWNER = 2 to make SI master controller the owner of the SI interface.
2. Write BAUDR = 250 to get 1 MHz baud rate.
3. Write SER = 2 to use SI_nCS1.
4. Write CTRLR0 = 0x10F for 16-bit data frame and transmit only.
5. Write SIMCEN = 1 to enable the SI master controller.
6. Write DR[0] = 0x8000, 0x0081, 0x8181, 0x8100 to configure SI slave for big endian / most significant bit first mode.
7. Wait for SR.TFE = 1 and SR.BUSY = 0 for chip select to deassert between accesses to the SI slave.
8. Write DR[0] = 0x8040, 0x0101, 0x2345, 0x6700 to write DEVCPU_GCB::GPR in slave device.
9. Wait for SR.TFE = 1 and SR.BUSY = 0, then disable the SI master controller by writing SIMCEN = 0.

When reading from the SI slave, CTRLR0.TMOD must be configured for transmit and receive. One-byte padding is appropriate for a 1 MHz baud rate.

3.8.3 DDR3/DDR3L MEMORY CONTROLLER

This section provides information about how to configure and initialize the DDR3/DDR3L memory controller.

The following table lists the registers associated with the memory controller.

TABLE 3-21: DDR3/DDR3L CONTROLLER REGISTERS

Register	Description
ICPU_CFG::RESET	Reset Control
ICPU_CFG::MEMCTRL_CTRL	Control
ICPU_CFG::MEMCTRL_CFG	Configuration
ICPU_CFG::MEMCTRL_STAT	Status
ICPU_CFG::MEMCTRL_REF_PERIOD	Refresh period configuration
ICPU_CFG::MEMCTRL_ZQCAL	DDR3/DDR3L ZQ-calibration
ICPU_CFG::MEMCTRL_TIMING0	Timing configuration
ICPU_CFG::MEMCTRL_TIMING1	Timing configuration
ICPU_CFG::MEMCTRL_TIMING2	Timing configuration
ICPU_CFG::MEMCTRL_TIMING3	Timing configuration
ICPU_CFG::MEMCTRL_MR0_VAL	Mode register 0 value
ICPU_CFG::MEMCTRL_MR1_VAL	Mode register 1 value
ICPU_CFG::MEMCTRL_MR2_VAL	Mode register 2 value
ICPU_CFG::MEMCTRL_MR3_VAL	Mode register 3 value
ICPU_CFG::MEMCTRL_TERMRES_CTRL	Termination resistor control
ICPU_CFG::MEMCTRL_DQS_DLY	DQS window configuration
ICPU_CFG::MEMPHY_CFG	SSTL configuration

TABLE 3-21: DDR3/DDR3L CONTROLLER REGISTERS (CONTINUED)

Register	Description
ICPU_CFG::MEMPHY_ZCAL	SSTL calibration

The memory controller works with JEDEC-compliant DDR3/DDR3L memory modules. The controller runs 312.50 MHz and supports one or two byte lanes in either single or dual row configurations (one or two chip selects). The controller supports up to 16 address bits. The maximum amount of physical memory that can be attached to the controller is 1 gigabyte.

The memory controller supports ECC encoding/decoding by using one of the two lanes for ECC information. Enabling ECC requires that both byte lanes are populated. When ECC is enabled, half the total physical memory is available for software use; the other half is used for ECC information.

The following steps are required to bring up the memory controller.

1. Release the controller from reset by clearing RESET.MEM_RST_FORCE.
2. Configure timing and mode parameters. These depend on the DDR3/DDR3L module and configuration selected for the device. For more information, see Section 3.8.3.1, DDR3/DDR3L Timing and Mode Configuration Parameters.
3. Enable and calibrate the SSTL I/Os. For more information, see Enabling and Calibrating SSTL I/Os.
4. Initialize the memory controller and modules. For more information, see Memory Controller and Module Initialization.
5. Calibrate the DQS read window. For more information, see DQS Read Window Calibration.
6. Optionally configure ECC and 1 gigabyte support. For more information see ECC and 1 Gigabyte Support.

Note For selected DDR3/DDR3L modules, the bring-up of the memory controller is already implemented as part of the Board Support Package (BSP). For more information about implementing the bring-up procedure, see Microsemi eCos BSP and Tool Chain, which is available on the Microchip Web site at www.Microsemi.com.

3.8.3.1 DDR3/DDR3L Timing and Mode Configuration Parameters

This section provides information about each of the parameters that must be configured prior to initialization of the memory controller. It provides a quick overview of fields that must be configured and their recommended values. For more information about each field, see the register list.

The memory controller operates at 312.5 MHz, and the corresponding DDR clock period is 3.2 ns (DDR625). When a specific timing value is used in this section (for example, t_{MOD}), it means that the corresponding value from the memory module datasheet is expressed in memory controller clock cycles (divided by 3.2 ns and rounded up and possibly adjusted for minimum clock cycle count).

The following table defines general parameters for DDR3/DDR3L required when configuring the memory controller.

TABLE 3-22: GENERAL DDR3/DDR3L TIMING AND MODE PARAMETERS

Parameter	DDR3/DDR3L
RL	CL
WL	CWL
MD	t_{MOD}
ID	t_{XPR}
SD	t_{DLLK}
OW	2
OR	2
RP	t_{RP}
FAW	t_{FAW}
BL	4
MR0	((RL-4)<<4) | ((t_{WR-4})<<9)
MR1	0x0040
MR2	((WL-5)<<3)
MR3	0x0000

The following table lists the memory controller registers that must be configured before initialization. It is recommended to configure all fields listed, even when ODT or dual row configurations are not used.

TABLE 3-23: MEMORY CONTROLLER CONFIGURATION

Register Field	Description
MEMCTRL_CFG.DDR_MODE	Set to 1 for DDR3/DDR3L mode.
MEMCTRL_CFG.DDR_WIDTH	Set to 0 when using 1 byte lane. Set to 1 when using 2 byte lanes.
MEMCTRL_CFG.MSB_COL_ADDR	Set to one less than the number of column address bits for the modules.
MEMCTRL_CFG.MSB_ROW_ADDR	Set to one less than the number of row address bits for the modules. See field description for example.
MEMCTRL_CFG.BANK_CNT	Set to 1.
MEMCTRL_CFG.BURST_LEN	Set to 1.
MEMCTRL_CFG.BURST_SIZE	Set to same value as MEMCTRL_CFG.DDR_WIDTH.
MEMCTRL_CFG.DDR_ECC_ENA	Set to 0 during initialization of the memory controller.
MEMCTRL_CFG.DDR_ECC_COR_ENA	Set to 0 during initialization of the memory controller.
MEMCTRL_CFG.DDR_ECC_ERR_ENA	Set to 0 during initialization of the memory controller.
MEMCTRL_CFG.DDR_512MBYTE_PLUS	Set to 0 during initialization of the memory controller.
MEMCTRL_REF_PERIOD.REF_PERIOD	Set to (tREFI(ns)/3.2ns) round down to the nearest integer.
MEMCTRL_REF_PERIOD.MAX_PEND_REF	Set to 8.
MEMCTRL_TIMING0.RD_DATA_XFR_DLY	Set to (RL-3).
MEMCTRL_TIMING0.WR_DATA_XFR_DLY	Set to (WL-1).
MEMCTRL_TIMING0.RD_TO_PRECH_DLY	Set to (BL-1).
MEMCTRL_TIMING0.WR_TO_PRECH_DLY	Set to (WL+BL+tWR-1).
MEMCTRL_TIMING0.RAS_TO_PRECH_DLY	Set to (tRAS(min)-1).
MEMCTRL_TIMING0.RD_CS_CHANGE_DLY	Set to BL.
MEMCTRL_TIMING0.WR_CS_CHANGE_DLY	Set to (BL-1).
MEMCTRL_TIMING0.RD_TO_WR_DLY	Set to (RL+BL+1-WL).
MEMCTRL_TIMING1.WR_TO_RD_DLY	Set to (WL+BL+tWTR-1).
MEMCTRL_TIMING1.RAS_TO_CAS_DLY	Set to (tRCD-1).
MEMCTRL_TIMING1.RAS_TO_RAS_DLY	Set to (tRRD-1).
MEMCTRL_TIMING1.PRECH_TO_RAS_DLY	Set to (tRP-1).
MEMCTRL_TIMING1.BANK8_FAW_DLY	Set to (FAW-1).
MEMCTRL_TIMING1.RAS_TO_RAS_SAME_BANK_DLY	Set to (tRC-1).
MEMCTRL_TIMING2.INIT_DLY	Set to (ID-1) during initialization of the memory controller.
MEMCTRL_TIMING2.REF_DLY	Set to (tRFC-1).
MEMCTRL_TIMING2.MDSET_DLY	Set to (MD-1).
MEMCTRL_TIMING2.PRECH_ALL_DLY	Set to (RP-1).
MEMCTRL_TIMING3.WR_TO_RD_CS_CHANGE_DLY	Set to (WL+tWTR-1).

TABLE 3-23: MEMORY CONTROLLER CONFIGURATION (CONTINUED)

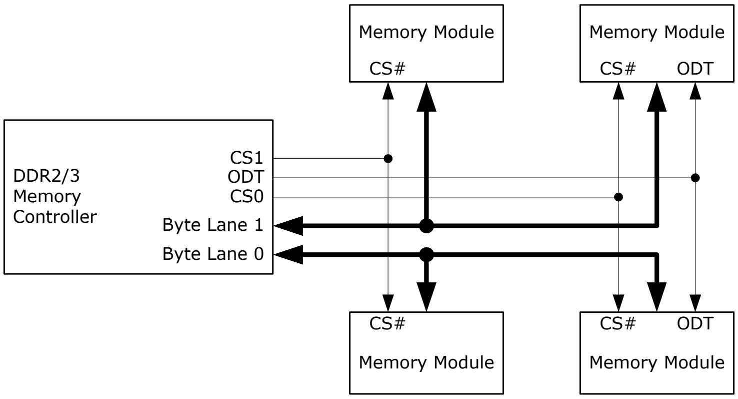
Register Field	Description
MEMCTRL_TIMING3.LOCAL_ODT_RD_DLY	Set to (RL-3).
MEMCTRL_TIMING3.ODT_WR_DLY	Set to (OW-1).
MEMCTRL_TIMING3.ODT_RD_DLY	Set to (OR-1).
MEMCTRL_MR0_VAL.MR0_VAL	Set to MR0.
MEMCTRL_MR1_VAL.MR1_VAL	Set to MR1.
MEMCTRL_MR2_VAL.MR2_VAL	Set to MR2.
MEMCTRL_MR3_VAL.MR3_VAL	Set to MR3.

The memory controller supports two chip selects and automatically selects modules based on the configured address width and bank count. The resulting pool of memory is one large consecutive region.

3.8.3.2 On-Die Termination

An end termination scheme must be configured when writing from the device to the memory modules, the memory module furthest away from the device must be connected to the ODT output. Enable assertion of ODT output for all write access by setting MEMCTRL_TERMRES_CTRL = 0x0000000C.

FIGURE 3-22: MEMORY CONTROLLER ODT HOOKUP



3.8.3.3 Enabling and Calibrating SSTL I/Os

Prior to controller initialization, the device’s SSTL I/O drivers must be enabled and calibrated to correct drive strength and termination resistor values. It is recommended that the device’s I/O drive and termination strengths be 60 Ω/60 Ω.

Complete the following tasks to enable and calibrate the SSTL I/Os.

1. Enable the SSTL mode by clearing MEMPHY_CFG.PHY_RST and setting MEMPHY_CFG.PHY_SSTL_ENA.
2. Perform calibration with the previously mentioned drive and termination strengths by writing 0xEF to MEMPHY_ZCAL.
3. Wait until MEMPHY_ZCAL.ZCAL_ENA is cleared (indicates calibration is done).
4. Enable drive of the SSTL I/Os by setting MEMPHY_CFG.PHY_CK_OE, MEMPHY_CFG.PHY_CL_OE, and MEMPHY_CFG.PHY_ODT_OE.

The SSTL interface is now enabled and calibrated, and the initialization of the memory controller can commence.

3.8.3.4 Memory Controller and Module Initialization

After memory controller parameters are configured, and after the SSTL I/Os are enabled and calibrated, the memory controller and modules are initialized by setting MEMCTRL_CTRL.INITIALIZE.

During initialization, the memory controller automatically follows the proper JEDEC defined procedure for initialization and writing of mode registers to the DDR3/DDR3L memory modules.

The memory controller automatically clears MEMCTRL_CTRL.INITIALIZE and sets the MEMCTRL_STAT.INIT_DONE field after the controller and the memory module (or modules) are operational. Software must wait for the INIT_DONE indication before continuing to calibrate the read window.

3.8.3.5 DQS Read Window Calibration

After initialization of the memory controller, successful writes to the memory are guaranteed. Reading is not yet possible, because the round trip delay between controller and memory modules are not calibrated.

Calibration of the read window includes writing a known value to the start of the DDR memory and then continually reading this value while adjusting the DQS window until the correct value is read from the memory. The calibration procedure depends on the number of byte lanes that are used (see MEMCTRL_CFG.DDR_WIDTH).

Note For an ECC system, the calibration procedure for two byte lanes must be used, because the memory controller is initialized without ECC enabled.

Calibration of One Byte Lane

The following procedure applies to systems where MEMCTRL_CFG.DDR_WIDTH is 0:

- Write 0x000000FF to SBA address 0x20000000.
- Set the MEMCTRL_DQS_DLY[0].DQS_DLY field to 0.

the DQS_DLY field beyond its maximum value. If the DQS_DLY maximum value is exceeded, it is an indication something is incorrect, and the memory controller will not be operational.

1. Read byte address 0x20000000. If the content is different from 0xFF, increment MEMCTRL_DQS_DLY[0].DQS_DLY by one, and repeat step 1, else continue to step 2.
2. Read byte address 0x20000000. If the content is different from 0x00, increment MEMCTRL_DQS_DLY[0].DQS_DLY by one, and repeat step 2, else continue to step 3.
3. Decrement MEMCTRL_DQS_DLY[0].DQS_DLY by three.

After the last step, which configures the appropriate DQS read window, the memory is ready for random read and write operations. For more information about ECC or 1 gigabyte support, see Section 3.8.3.6, ECC and 1-Gigabyte Support.

Calibration of Two Byte Lanes

This procedure applies to systems where MEMCTRL_CFG.DDR_WIDTH is 1:

- Write 0x0000FFFF to SBA address 0x20000000.
- Write 0x00000000 to SBA address 0x20000004.
- Set the MEMCTRL_DQS_DLY[0].DQS_DLY field to 0.
- Set the MEMCTRL_DQS_DLY[1].DQS_DLY field to 0.

Perform the following steps to calibrate the read window. Do not increment the DQS_DLY fields beyond their maximum value. If a DQS_DLY maximum value is exceeded, it is an indication something is incorrect, and the memory controller will not be operational.

1. Read byte addresses 0x20000000 and 0x20000001 from memory.

   If the contents of byte address 0x20000000 is different from 0xFF, increment MEMCTRL_DQS_DLY[0].DQS_DLY by one.

   If the contents of byte address 0x20000001 is different from 0xFF, increment MEMCTRL_DQS_DLY[1].DQS_DLY by one.

   If any DQS_DLY was incremented repeat step 1, else continue to step 2.
2. Read byte addresses 0x20000000 and 0x20000001 from memory.

   If the contents of byte address 0x20000000 is different from 0x00, increment MEMCTRL_DQS_DLY[0].DQS_DLY by one.

   If the contents of byte address 0x20000001 is different from 0x00, increment MEMCTRL_DQS_DLY[1].DQS_DLY by one.

   If any DQS_DLY was incremented repeat step 2, else continue to step 3.
3. Decrement MEMCTRL_DQS_DLY[0].DQS_DLY by three, and decrement MEMCTRL_DQS_DLY[1] by three.

The last step configures the appropriate DQS read windows. After this step the memory is ready for random read and write operations. If ECC or 1 gigabyte is needed, see ECC and 1 Gigabyte Support.

3.8.3.6 ECC and 1-Gigabyte Support

The memory controller supports ECC, allowing correction of two bits and detection of three-bit errors in every byte that is read from memory.

When ECC is enabled, one byte lane is used for data words and the other is used for ECC check-word. After enabling ECC, the amount of memory available to software is half of the physical memory (the lower half of the memory region).

ECC support must be enabled just after read calibration by setting MEMCTRL_CFG.DDR_ECC_ENA = 1 and at the same time setting MEMCTRL_CFG.BURST_SIZE = 0.

When ECC is enabled, the memory controller automatically corrects two-bit errors and detects three-bit errors. It is possible to propagate these events to the SBA to trigger a bus-error event in the VCore-III CPU. Non-correctable error indications are forwarded when MEMCTRL_CFG.DDR_ECC_ERR_ENA = 1. Correctable “errors” are forwarded when MEMCTRL_CFG.DDR_ECC_COR_ENA = 1. The VCore-III CPU can generate interrupt on these events so that software can take the appropriate action.

The memory controller supports up to 1 gigabyte of physical memory. To access more than 512 megabytes from the VCore-III CPU, set MEMCTRL_CFG.DDR_512MBYTE_PLUS = 1 to change the SBA memory map. For more information, see Shared Bus. When ECC is enabled with 1 gigabyte of physical memory, MEMCTRL_CFG.DDR_512MBYTE_PLUS does not have to be set, because only half of the physical memory is available to software.

3.8.4 TIMERS

This section provides information about the timers. The following table lists the registers associated with timers.

TABLE 3-24: TIMER REGISTERS

Register	Description
ICPU_CFG::TIMER_CTRL	Enable/disable timer
ICPU_CFG::TIMER_VALUE	Current timer value
ICPU_CFG::TIMER_RELOAD_VALUE	Value to load when wrapping
ICPU_CFG::TIMER_TICK_DIV	Common timer-tick divider

There are three decrementing 32-bit timers in the VCore-III system that run from a common divider. The common divider is driven by a fixed 250 MHz clock and can generate timer ticks from 0.1 µs (10 MHz) to 1 ms (1 kHz), configurable through TIMER_TICK_DIV. The default timer tick is 100 µs (10 kHz).

Software can access each timer value through the TIMER_VALUE[*n*] registers. These can be read or written at any time, even when the timers are active.

When a timer is enabled through TIMER_CTRL[*n*].TIMER_ENA, it decrements from the current value until it reaches zero. An attempt to decrement a TIMER_VALUE[*n*] of zero generates interrupt and assigns TIMER_VALUE[*n*] to the contents of TIMER_RELOAD_VALUE[*n*]. Interrupts generated by the timers are sent to the VCore-III interrupt controller. From here, interrupts can be forwarded to the VCore-III CPU or to an external CPU. For more information, see Interrupt Controller.

By setting TIMER_CTRL[*n*].ONE_SHOT_ENA, the timer disables itself after generating one interrupt. By default, timers will decrement, interrupt, and reload indefinitely (or until disabled by clearing TIMER_CTRL[*n*].TIMER_ENA).

A timer can be reloaded from TIMER_RELOAD_VALUE[*n*] at the same time as it is enabled by setting TIMER_CTRL[*n*].FORCE_RELOAD and TIMER_CTRL[*n*].TIMER_ENA at the same time.

Example Configure Timer0 to interrupt every 1 ms. With the default timer tick of 100 µs ten timer ticks are needed for a timer that wraps every 1 ms. Configure TIMER_RELOAD_VALUE[0] to 0x9, then enable the timer and force a reload by setting TIMER_CTRL[0].TIMER_ENA and TIMER_CTRL[0].FORCE_RELOAD at the same time.

3.8.5 UARTS

This section provides information about the UART (Universal Asynchronous Receiver/Transmitter) controllers. There are two independent UART controller instances in the VCore-III system: UART and UART2. These instances are identical copies and anywhere in this description the word UART can be replaced by UART2.

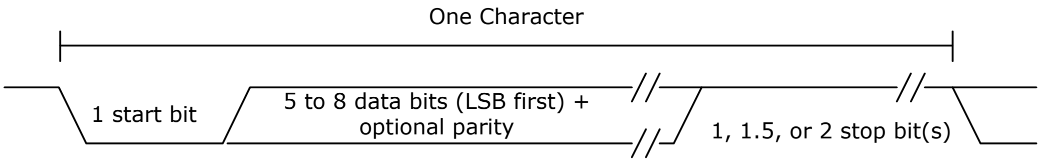
The following table lists the registers associated with the UART.

TABLE 3-25: UART REGISTERS

Register	Description
UART::RBR_THR	Receive buffer/transmit buffer/Divisor (low)
UART::IER	Interrupt enable/divisor (high)
UART::IIR_FCR	Interrupt identification/FIFO control
UART::LCR	Line control
UART::MCR	Modem control
UART::LSR	Line status
UART::MSR	Modem status
UART::SCR	Scratchpad
UART::USR	UART status

The VCore-III system UART is functionally based on the industry-standard 16550 UART (RS232 protocol). This implementation features a 16-byte receive and a 16-byte transmit FIFO.

FIGURE 3-23: UART TIMING



The number of data bits, parity, parity-polarity, and stop-bit lengths are all programmable using LCR.

The UART pins on the device are overlaid functions on the GPIO interface. Before enabling the UART, the VCore-III CPU must enable overlaid modes for the appropriate GPIO pins. For more information about overlaid functions on the GPIOs for these signals, see Section 3.8.8.1, GPIO Overlaid Functions.

The following table lists the pins of the UART interfaces.

TABLE 3-26: UART INTERFACE PINS

Pin Name	I/O	Description
UART_RXD/GPIO	I	UART receive data
UART_TXD/GPIO	O	UART transmit data
UART2_RXD/GPIO	I	UART2 receive data
UART2_TXD/GPIO	O	UART2 transmit data

The baud rate of the UART is derived from the VCore-III system frequency. The divider value is indirectly set through the RBR_THR and IER registers. The baud rate is equal to the VCore-III system clock frequency divided by sixteen times the value of the baud rate divisor. A divider of zero disables the baud rate generator and no serial communications occur. The default value for the divisor register is zero.

Example Configuring a baud rate of 9600 in a 250 MHz VCore-III system. To generate a baud rate of 9600, the divisor register must be set to 0x65C (250 MHz/(16 × 9600 Hz)). Set LCR.DLAB and write 0x5C to RBR_THR and 0x06 to IER (this assumes that the UART is not in use). Finally, clear LCR.DLAB to change the RBR_THR and IER registers back to the normal mode.

By default, the FIFO mode of the UART is disabled. Enabling the 16-byte receive and 16-byte transmit FIFOs (through IIR_FCR) is recommended.

Note: Although the UART itself supports RTS and CTS, these signals are not available on the pins of the device.

3.8.5.1 UART Interrupt

The UART can generate interrupt whenever any of the following prioritized events are enabled (through IER).

- Receiver error
- Receiver data available
- Character timeout (in FIFO mode only)
- Transmit FIFO empty or at or below threshold (in programmable THRE interrupt mode)

When an interrupt occurs, the IIR_FCR register can be accessed to determine the source of the interrupt. Note that the IIR_FCR register has different purposes when reading or writing. When reading, the interrupt status is available in bits 0 through 3. For more information about interrupts and how to handle them, see the IIR_FCR register description.

Example Enabling interrupt when transmit fifo is below one-quarter full. To get this type of interrupt, the THRE interrupt must be used. First, configure TX FIFO interrupt level to one-quarter full by setting IIR_FCR.TET to 10; at the same time, ensure that the IIR_FCR.FIFOE field is also set. Set IER.PTIME to enable the THRE interrupt in the UART. In addition, the VCore-III interrupt controller must be configured for the CPU to be interrupted. For more information, see Interrupt Controller.

3.8.6 TWO-WIRE SERIAL INTERFACE

This section provides information about the two-wire serial interface controller in the VCore-III system. The following table lists the registers associated with the two-wire serial interface.

TABLE 3-27: TWO-WIRE SERIAL INTERFACE REGISTERS

Register	Description
TWI::CFG	General configuration.
TWI::TAR	Target address.
TWI::SAR	Slave address.
TWI::DATA_CMD	Receive/transmit buffer and command.
TWI::SS_SCL_HCNT	Standard speed high time clock divider.
TWI::SS_SCL_LCNT	Standard speed low time clock divider.
TWI::FS_SCL_HCNT	Fast speed high time clock divider.
TWI::FS_SCL_LCNT	Fast speed low time clock divider.
TWI::INTR_STAT	Masked interrupt status.
TWI::INTR_MASK	Interrupt mask register.
TWI::RAW_INTR_STAT	Unmasked interrupt status.
TWI::RX_TL	Receive FIFO threshold for RX_FULL interrupt.
TWI::TX_TL	Transmit FIFO threshold for TX_EMPTY interrupt.
TWI::CLR_*	Individual CLR_* registers are used for clearing specific interrupts. See register descriptions for corresponding interrupts.
TWI::CTRL	Control register.
TWI::STAT	Status register.
TWI::TXFLR	Current transmit FIFO level.
TWI::RXFLR	Current receive FIFO level.
TWI::TX_ABRT_SOURCE	Arbitration sources.
TWI::SDA_SETUP	Data delay clock divider.
TWI::ACK_GEN_CALL	Acknowledge of general call.
TWI::ENABLE_STATUS	General two-wire serial controller status.
ICPU_CFG::TWI_CONFIG	Configuration of TWI_SDA hold-delay.
ICPU_CFG::TWI_SPIKE_FILTER_CFG	Configuration of TWI_SDA spike filter.

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 masters, as well as both 7-bit and 10-bit addressing, are also supported.

The two-wire serial interface controller operates as master 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 VCore-III CPU must enable overlaid functions for the appropriate GPIO pins. For more information about overlaid functions on the GPIOs for these signals, see Section 3.8.8.1, GPIO Overlaid Functions.

The following table lists the pins of the two-wire serial interface.

TABLE 3-28: TWO-WIRE SERIAL INTERFACE PINS

Pin Name	I/O	Description
TWI_SCL/GPIO	I/O	Two-wire serial interface clock, open-collector output.
TWI_SDA/GPIO	I/O	Two-wire serial interface data, open-collector output.
TWI_SCL_M*n*/GPIO	I/O	Two-wire serial interface multiplexed clocks (12 instances in total), open-collector outputs.
TWI_SCL_M*n*_AD/ GPIO	I/O	Two-wire serial interface multiplexed clocks (4 instances in total). Same as TWI_SCL_M*n*/GPIO but the pin is always driven meaning feedback from slave devices is not possible.

Setting CTRL.ENABLE enables the controller. The controller can be disabled by clearing the CTRL.ENABLE field, there is a chance that disabling is not allowed (at the time when it is attempted); the ENABLE_STATUS register shows if the controller was successful disabled.

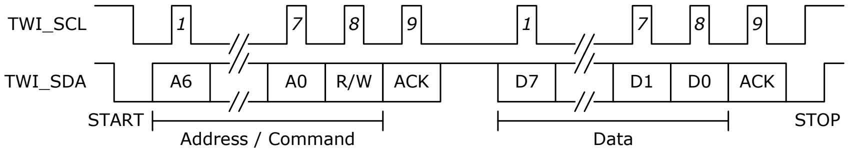
Before enabling the controller, the user must decide on either standard or fast mode (CFG.SPEED) and configure clock dividers for generating the correct timing (SS_SCL_HCNT, SS_SCL_LCNT, FS_SCL_HCNT, FS_SCL_LCNT, and SDA_SETUP). The configuration of the divider registers depends on the VCore-III system clock frequency. The register descriptions explain how to calculate the required values.

Some two-wire serial devices require a hold time on TWI_SDA after TWI_SCL when transmitting from the two-wire serial interface controller. The device supports a configurable hold delay through the TWI_CONFIG register.

The two-wire serial interface controller has an 8-byte combined receive and transmit FIFO.

During normal operation of the two-wire serial interface controller, the STATUS register shows the activity and FIFO states.

FIGURE 3-24: TWO-WIRE SERIAL INTERFACE TIMING FOR 7-BIT ADDRESS ACCESS



3.8.6.1 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 TAR.GC_OR_START_ENA or TAR.GC_OR_START. The target/slave address is configured using the TAR register.

3.8.6.2 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 RAW_INTR_STAT register contains interrupt fields that are always set when their trigger conditions occur. The INTR_MASK register is used for masking interrupts and allowing interrupts to propagate to the INTR_STAT register. When set in the INTR_STAT register, the two-wire serial interface controller asserts interrupt toward the VCore-III interrupt controller.

The 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 CLR_* register (for example, CLR_RX_OVER) clears the interrupt.

3.8.6.3 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_M*n*) for the two-wire serial devices rather than TWI_SCL. Depending on which device or devices it needs to talk to, software can then enable/disable the various clocks.

From the two-wire serial controller's point of view, it does not know if it is using TWI_SCL or TWI_SCL_M*n* clocks. When using multiplexed mode, software needs to enable/disable individual TWI_SCL_M*n* 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_M*n* lines that are enabled.

To enable multiplexed clocks, configure the TWI_SCL_Mn overlaid mode in the GPIO controller during initialization. Individual TWI_SCL_M*n* clocks are then enabled, when needed, 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 TWI_SCL_M*n* clocks are not driven during access and the feedback is disabled.

Note In multiprocessor systems, the DEVCPU_GCB::GPIO_OUT_SET and DEVCPU_GCB::GPIO_OUT_CLR registers can be used to avoid race conditions.

3.8.7 MII MANAGEMENT CONTROLLER

This section provides information about the MII Management (MIIM) controllers. The following table lists the registers associated with the MII Management controllers.

TABLE 3-29: 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 device contains 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 first MIIM controller is connected to the internal PHYs of the device.

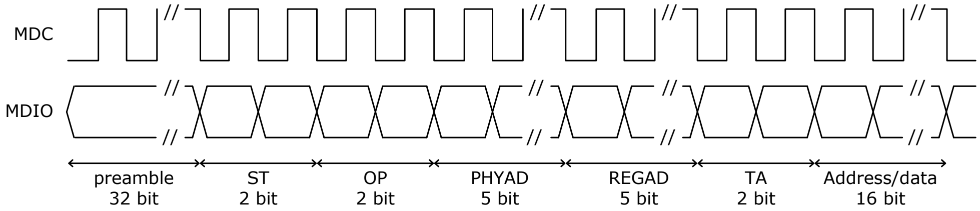
The MIIM interface pins for the second controller are overlaid functions on the GPIO interface. Before using this MIIM controller, the overlaid functions for the appropriate GPIO pins must first be enabled. For more information about overlaid functions on the GPIOs for these signals, see Section 3.8.8.1, GPIO Overlaid Functions.

TABLE 3-30: MIIM MANAGEMENT CONTROLLER PINS

Pin Name	I/O	Description
MDC1/GPIO	O	MIIM clock for controller 1
MDIO1/GPIO	I/O	MIIM data input/output for controller 1

The MDIO signal is changed or sampled on the falling edge of the MDC clock by the controller. The MDIO pin is tristated in-between access and when expecting read data.

FIGURE 3-25: MII MANAGEMENT TIMING



3.8.7.1 Clock Configuration

The frequency of the management interface clock generated by the MIIM controller is derived from the switch core's 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. The maximum frequency of the MIIM clock is 25 MHz.

3.8.7.2 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.

3.8.7.3 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).

3.8.7.4 MII Management Interrupt

The MII management controllers can generate interrupts during PHY scanning. Each MII management controller has a separate interrupt signal to the interrupt controller. Interrupt is asserted when one or more PHYs have a mismatch during scan. The interrupt is cleared by reading the MII_SCAN_RSLTS_STICKY register, which resets all MII_SCAN_RSLTS_STICKY indications.

3.8.8 GPIO CONTROLLER

This section provides information about the use of GPIO pins.

The following table lists the registers associated with GPIO.

TABLE 3-31: 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_MAP	Mapping of parallel signal detects

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.

3.8.8.1 GPIO Overlaid Functions

Most of the GPIO pins have overlaid (alternative) functions that can be enabled through replicated GPIO_ALT registers.

To enable a particular GPIO[*n*] pin with the alternate function, set the G_ALT[*n*] register field in the replicated registers as follows:

- Overlaid mode 1, set GPIO_ALT[1][*n*], GPIO_ALT[0][*n*] = 1.
- Overlaid mode 2, set GPIO_ALT[1][*n*], GPIO_ALT[0][*n*] = 2.
- Normal GPIO mode, set GPIO_ALT[1][*n*], GPIO_ALT[0][*n*] = 0.

When the MIIM slave mode is enabled through the VCore_CFG strapping pins, specific GPIO pins are overtaken and used for the MIIM slave interface.

During reset, the VCore_CFG interface is sampled and used for VCore configuration. After reset, the device is released, and the GPIOs can be used for output or inputs. For more information, see Section 3.1, VCore-III Configurations.

The following table maps the GPIO pins and available overlaid functions.

TABLE 3-32: GPIO OVERLAID FUNCTIONS

Name	Overlaid Function 1	Overlaid Function 2	Overlaid Function 3	Configuration or Interface
GPIO_0	SG0_CLK			REFCLK0_CONF0
GPIO_1	SG0_DO			REFCLK0_CONF1
GPIO_2	SG0_DI			
GPIO_3	SG0_LD			REFCLK0_CONF2
GPIO_4	IRQ0_IN	IRQ0_OUT	TWI_SCL_M13	
GPIO_5	IRQ1_IN	IRQ1_OUT	PCI_Wake	
GPIO_6	UART_RXD	TWI_SCL_M0		
GPIO_7	UART_TXD	TWI_SCL_M1		
GPIO_8	SI_nCS1	TWI_SCL_M2	IRQ0_OUT	
GPIO_9	SI_nCS2	TWI_SCL_M3	IRQ1_OUT	
GPIO_10	PTP_2	TWI_SCL_M4	SFP0_SD	
GPIO_11	PTP_3	TWI_SCL_M5	SFP1_SD	
GPIO_12	UART2_RxD	TWI_SCL_M6	SFP2_SD	
GPIO_13	UART2_TxD	TWI_SCL_M7	SFP3_SD	
GPIO_14	MIIM1_MDC	TWI_SCL_M8	SFP4_SD	MIIM_SLV_MDC
GPIO_15	MIIM1_MDIO	TWI_SCL_M9	SFP5_SD	MIIM_SLV_MDIO
GPIO_16	TWI_SDA		SI_nCS3	
GPIO_17	TWI_SCL	TWI_SCL_M10	SI_nCS4	
GPIO_18	PTP_0	TWI_SCL_M11_AD		VCORE_CFG0
GPIO_19	PTP_1	TWI_SCL_M12_AD		VCORE_CFG1
GPIO_20	RECO_CLK0	TACHO	TWI_SCL_M14_AD	VCORE_CFG2
GPIO_21	RECO_CLK1	PWM	TWI_SCL_M15_AD	VCORE_CFG3

3.8.8.2 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 VCore-III 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.

3.8.8.3 Parallel Signal Detect

The GPIO controller has 6 programmable parallel signal detects, SFP0_SD through SFP5_SD. When parallel signal detect is enabled for a front port 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 port index from the serial GPIO controller that must be overwritten by setting GPIO_SD_MAP[*n*].G_SD_MAP, then enable the SFP*n*_SD function on the GPIOs.

The following table lists parallel signal detect pins, which are overlaid on GPIOs. For more information about overlaid functions on the GPIOs for these signals, see Section 3.8.8.1, GPIO Overlaid Functions.

TABLE 3-33: PARALLEL SIGNAL DETECT PINS

Register	I/O	Description
SFP0_SD/GPIO	I	Parallel signal detect 0
SFP1_SD/GPIO	I	Parallel signal detect 1
SFP2_SD/GPIO	I	Parallel signal detect 2

TABLE 3-33: PARALLEL SIGNAL DETECT PINS (CONTINUED)

Register	I/O	Description
SFP3_SD/GPIO	I	Parallel signal detect 3
SFP4_SD/GPIO	I	Parallel signal detect 4
SFP5_SD/GPIO	I	Parallel signal detect 5

3.8.9 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.

Each SIO controller supports up to 128 serial GPIOs (SGPIOs) organized into 32 ports, with four SGPIOs per port.

The following table lists the registers associated with the SIO controller.

TABLE 3-34: 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_TRIGGER0	Interrupt trigger mode 0 configuration
DEVCPU_GCB::SIO_INTR_TRIGGER1	Interrupt trigger mode 1 configuration
DEVCPU_GCB::SIO_INTR	Currently interrupting SGPIOs
DEVCPU_GCB::SIO_INTR_ENA	Interrupt enable
DEVCPU_GCB::SIO_INTR_IDENT	Currently active interrupts

The following table lists the SIO controller pins, which are overlaid functions on GPIO pins. For more information about overlaid functions on the GPIOs for these signals, see Section 3.8.8.1, GPIO Overlaid Functions.

TABLE 3-35: SIO CONTROLLER PINS

Pin Name	I/O	Description
SG0_CLK/GPIO	O	SIO clock output, frequency is configurable using SIO_CLOCK.SIO_CLK_FREQ.
SG0_DO/GPIO	O	SIO data output.
SG0_DI/GPIO	I	SIO data input.
SG0_LD/GPIO	O	SIO load data, polarity is configurable using SIO_CFG.SIO_LD_POLARITY.

The SIO controller works by shifting SGPIO values out on SG0_DO though a chain of shift registers on the PCB. After shifting a configurable number of SGPIO bits, the SIO controller asserts SG0_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 SG0_DO by sampling the SG0_DI input. The values sampled on SG0_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.

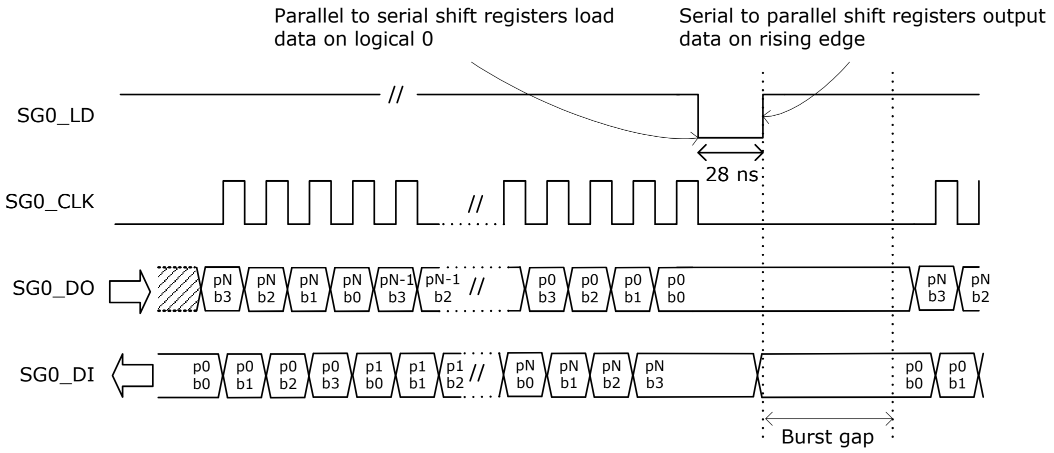
The number of shift registers in the chain is configurable. The SIO controller allows enabling of individual ports through SIO_PORT_ENA; only enabled ports are shifted out on SG0_DO. Ports that are not enabled are skipped during shifting of GPIO values.

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 SGPIOs 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, for example.

FIGURE 3-26: SIO TIMING



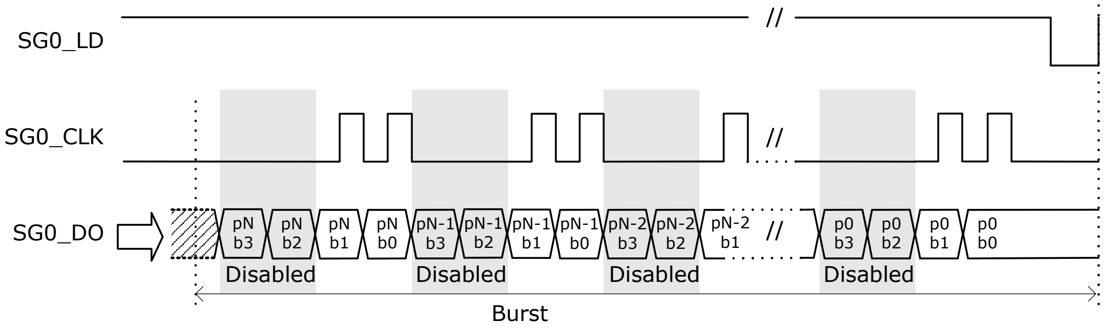
The SGPIO values are output in bursts followed by assertion of the SG0_LD signal. Values can be output as a single burst or as continuous bursts separated by a configurable burst gap. The maximum length of a burst is 32 × 4 data cycles. The burst gap is configurable in steps of approximately 1 ms, from 0 ms to 33 ms through SIO_CFG.SIO_BURST_GAP_DIS and SIO_CFG.SIO_BURST_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.

The following illustration shows what happens when the number of SGPIOs per port is configured to two (through SIO_CFG.SIO_PORT_WIDTH). Disabling ports (through SIO_PORT_ENA) is handled in the same way as disabling the SGPIO ports.

FIGURE 3-27: SIO TIMING WITH SGPIOS DISABLED



The frequency of the SG0_CLK clock output is configured through SIO_CLOCK.SIO_CLK_FREQ. The SG0_LD output is asserted after each burst; this output is asserted for a period of 25 ns to 30 ns. The polarity of SG0_LD is configured in SIO_CFG.SIO_LD_POLARITY.

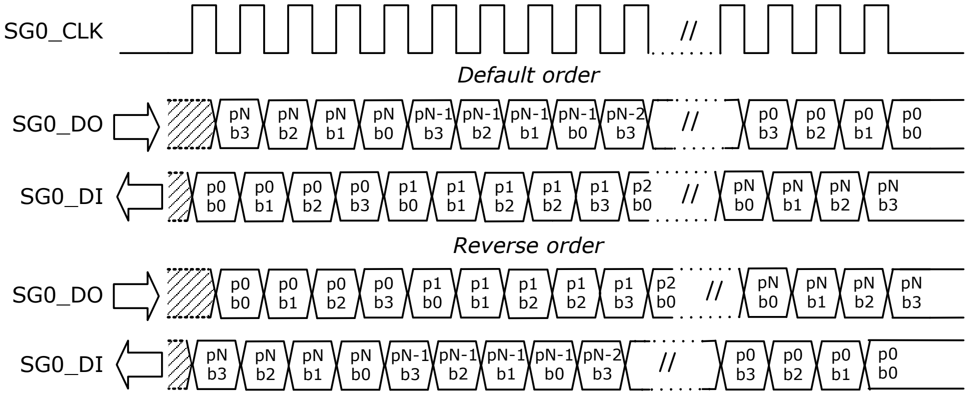
The SG0_LD output can be used to ensure that outputs are stable when serial data is being shifted through the registers. This can be done by using the SG0_LD output to shift the output values into serial-to-parallel registers after the burst is completed. If serial-to-parallel registers are not used, 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 SG0_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.

The following illustration shows the port order.

FIGURE 3-28: SGPIO OUTPUT ORDER



3.8.9.1 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).

Static Mode The static mode is used to assign a fixed value to the SGPIO, for example, fixed 0 or fixed 1.

Blink Mode 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 3-36: BLINK MODES

Register	Description
Blink Mode 0	0: 20 Hz blink frequency 1: 10 Hz blink frequency 2: 5 Hz blink frequency 3: 2.5 Hz blink frequency
Blink Mode 1	0: 20 Hz blink frequency 1: 10 Hz blink frequency 2: 5 Hz blink frequency 3: Burst toggle

Link Activity Mode The link activity mode makes the output blink when there is activity on the port module (Rx or Tx). The following table lists the mapping between the SIO port number and port modules.

TABLE 3-37: SIO CONTROLLER PORT MAPPING

SIO Port	SG0 Mapping
0	Port module 0
1	Port module 1
2	Port module 2
3	Port module 3
4	Port module 4
5	Port module 5
6	Port module 6
7	Port module 7
8	Port module 8
9	Port module 9
10	Port module 10
11	Free
12	Free
13	Free
14	Free
15	Free
16	Free
17	Free
18	Free
19	Free
20	Free
21	Free
22	Free
23	Free
24	Free
25	Free

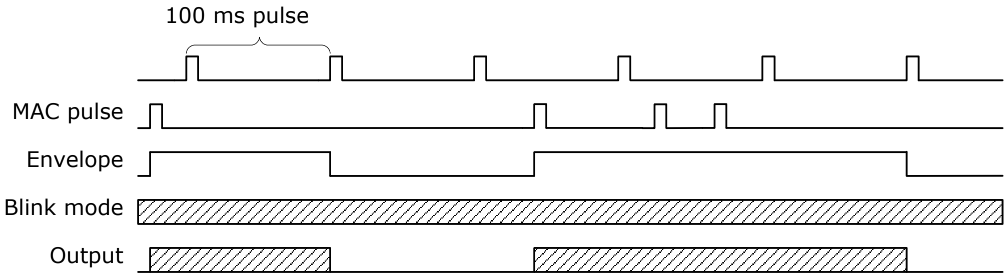
TABLE 3-37: SIO CONTROLLER PORT MAPPING (CONTINUED)

SIO Port	SG0 Mapping
26	Free
27	Free
28	Free
29	Free
30	Free
31	Free

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 de-asserted, 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.

The following illustration shows the link activity timing.

FIGURE 3-29: LINK ACTIVITY TIMING



3.8.9.2 SIO Interrupt

The SIO controller can generate interrupts based on the value of the input value of the SGPIOs. All interrupts are level sensitive.

Interrupts are enabled using the two registers. Interrupts can be individually enabled for each port in SIO_INTR_ENA (32 bits) and in SIO_CFG.SIO_GPIO_INTR_ENA (4 bits) interrupts are enabled for the four inputs per port. In other words, SIO_CFG.SIO_GPIO_INTR_ENA is common for all 32 ports.

The SIO controller has four interrupt registers that each has one bit for each of the 128 GPIOs:

- 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.
- 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.
- SIO_INTR_IDENT is the result of SIO_INTR with the disabled interrupts (from SIO_INTR_ENA and SIO_GPIO_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 and SIO_GPIO_INTR_ENA).

3.8.9.3 Loss of Signal Detection

The SIO controller can propagate loss of signal detection inputs directly to the signal detection input of the port modules. This is useful when, for example, SFP modules are connected to the device. The mapping between SIO ports and port modules is the same as for the link activity outputs; port 0 is connected to port module 0, port 1 is connected to port module 1, and so on.

The value of SGPIO bit 0 of each SIO port is forwarded directly to the loss of signal input on the corresponding port module. The port module must enable the loss of signal input locally.

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.

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.

3.8.10 FAN CONTROLLER

The device includes a fan controller that can be used to control and monitor a system fan. A pulse width modulation (PWM) output regulates the fan speed. The fan speed is monitored using a TACHO input. The fan controller is especially powerful when combined with the internal temperature sensor. For more information, see Section 3.8.11, Temperature Sensor.

The following table lists the registers associated with the fan controller.

TABLE 3-38: FAN CONTROLLER

Register	Description
DEVCPU_GCB::FAN_CFG	General configuration
DEVCPU_GCB::PWM_FREQ	Configuration of PWM frequency
DEVCPU_GCB::FAN_CNT	Fan revolutions counter

The following table lists fan controller pins, which are overlaid on GPIOs. For more information about overlaid functions on the GPIOs for these signals, see Section 3.8.8.1, GPIO Overlaid Functions.

TABLE 3-39: FAN CONTROLLER PINS

Register	I/O	Description
TACHO/GPIO	I	TACHO input for counting revolutions
PWM/GPIO	O	PWM fan output

The PWM output frequency depends on the value of DEVCPU_GCB::PWM_FREQ and the switch core clock frequency. If *pwm_freq* is the desired PWM frequency and *core_freq* is the switch core frequency, then DEVCPU_GCB::PWM_FREQ must be configured with value *core_freq/pwm_freq/256*. The DEVCPU_GCB::PWM_FREQ configuration register is 16bit wide and it defaults to value 24. The supported PWM frequency range depends on the core clock frequency. With 156.25MHz core frequency, the PWM can be configured in the range of 10Hz - 610kHz. The default PWM frequency is ~25kHz.

The low frequencies can be used for driving three-wire fans using a FET/transistor. The 25 kHz frequency can be used for four-wire fans that use the PWM input internally to control the fan. The duty cycle of the PWM output is programmable from 0% to 100%, with 8-bit accuracy. The polarity of the output can be controlled by FAN_CFG.INV_POL, so a duty-cycle of 100%, for example, can be either always low or always high.

The PWM output pin can be configured to act as a normal output or as an open-collector output, where the output value of the pin is kept low, but the output enable is toggled. The open-collector output mode is enabled by setting FAN_CFG.PWM_OPEN_COL_ENA.

Note: By using open-collector mode, it is possible to externally pull-up to a higher voltage than the maximum GPIO I/O supply. The GPIO pins are 3.3V-tolerable.

The speed of the fan is measured using a 16-bit counter that counts the rising edges on the TACHO input. A fan usually gives one to four pulses per revolution, depending on the fan type. The counter value is available in the FAN_CNT register. Depending on the value of FAN_CFG.FAN_STAT_CFG, the FAN_CNT register is updated in two different ways:

- If FAN_CFG.FAN_STAT_CFG is set, the FAN_CNT register behaves as a 16-bit wrapping counter that shows the total number of ticks on the TACHO input.
- If FAN_CFG.FAN_STAT_CFG is cleared, the FAN_CNT register is updated one time per second with the number of TACHO ticks received during the last second.

Optionally, the TACHO input is gated by the polarity-corrected PWM output by setting FAN_CFG.GATE_ENA, so that only TACHO pulses received while the polarity corrected PWM output is high are counted. Glitches on the TACHO input can occur right after the PWM output goes high. As a result, the gate signal is delayed by 10 µs when PWM goes high. There is no delay when PWM goes low, and the length of the delay is not configurable. Software must read the counter value in FAN_CNT and calculate the RPM of the fan.

An example of how to calculate the RPM of the fan is if the fan controller is configured to 100 Hz and a 20% duty cycle, each PWM pulse is high in 2 ms and low in 8 ms. If gating is enabled, the gating of the TACHO input is open in 1.99 ms and closed in 8.01 ms. If the fan is turning with 100 RPMs and gives two TACHO pulses per revolution, it will ideally give 200 pulses per minute. TACHO pulses are only counted in 19.99% of the time, so it will give 200 × 0.1999 = 39.98 pulses per minute. If the additional 10 µs gating time is ignored, the counter value is multiplied by 2.5 to get the RPM value, because there is a 20% duty cycle with two TACHO pulses per revolution. By multiplying with 2.5, the RPM value is calculated to 99.95, which is 0.05% off the correct value (due to the 10 µs gating time).

3.8.11 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.

The following table lists the registers associated with the temperature monitor.

TABLE 3-40: TEMPERATURE SENSOR REGISTERS

Register	Description
HSIO::TEMP_SENSOR_CTRL	Enabling of sensor
HSIO::TEMP_SENSOR_STAT	Temperature value

The temperature sensor is enabled by setting TEMP_SENSOR_CTRL.SAMPLE_ENA. After this the temperature sensor will sample temperature every 500 µs and show current temperature through TEMP_SENSOR_STATE.TEMP. The formula for converting TEMP field value to centigrade temperature is:

Temp (°C) = 177.4 – 0.8777 × TEMP_SENSOR_STATE.TEMP

It takes approximately 500 µs after setting SAMPLE_ENA until the first temperature sample is ready. The TEMP_SENSOR_SATE.TEMP_VALID field is set when the temperature value is available.

3.8.12 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 device has built-in protection from soft errors by using error correcting code (ECC) on critical memories. In addition, the device allows monitoring and reporting of soft error events.

The following table lists the registers associated with the memory integrity monitor.

TABLE 3-41: 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 by the ITGR software interrupt. For information about how to enable this interrupt, see Section 3.8.13, Interrupt Controller.

The memory integrity monitor operates in three different states: IDLE, LISTEN, and DETECT. After a reset, the monitor starts in the IDLE state.

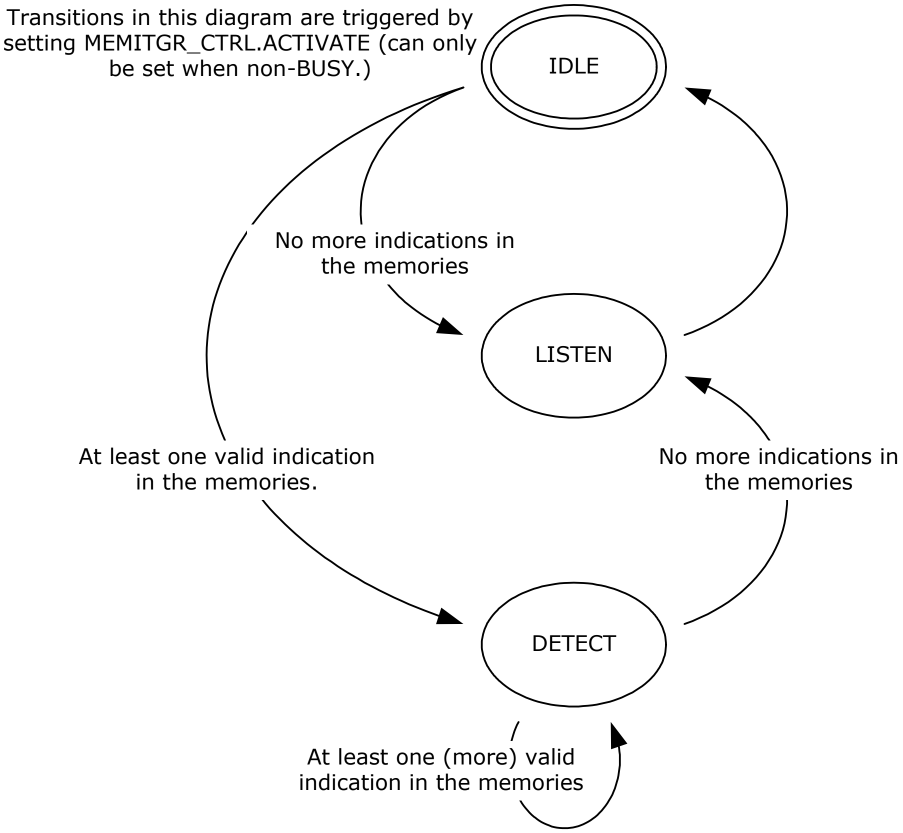
IDLE The monitor is deactivated and in quiet mode. In IDLE mode, the memories still correct, detect, and store indications locally, but they are not able to report indications to the monitor.

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 3-30: MONITOR STATE DIAGRAM


Transitions in this diagram are triggered by setting MEMITGR_CTRL.ACTIVATE (can only be set when non-BUSY.)
IDLE
No more indications in the memories
LISTEN
At least one valid indication in the memories.
No more indications in the memories
DETECT
At least one (more) valid indication in the memories

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.

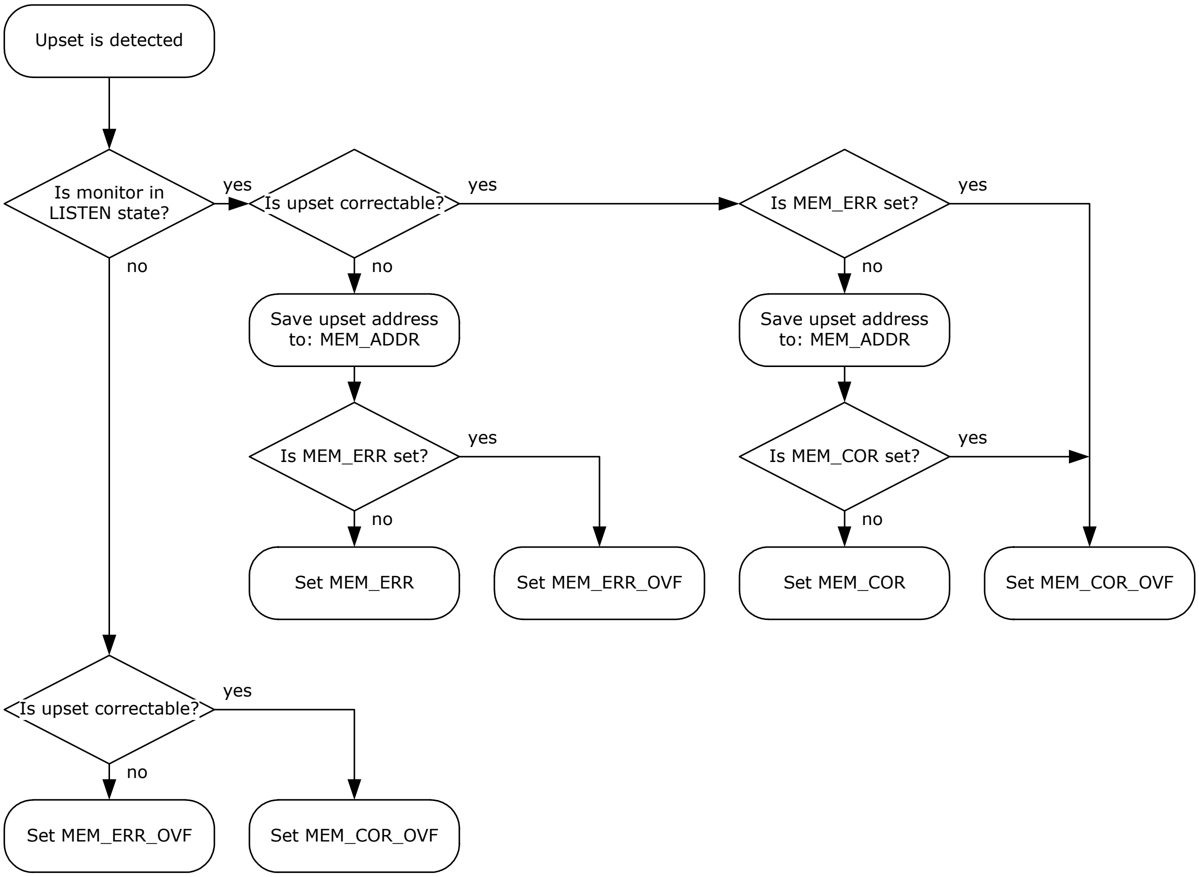
Before setting MEMITGR_CTRl.ACTIVATE for the first time, speed up the monitor by setting MEMITGR_DIV.MEM_DIV to the value specified in the registers list. The memories of the PCIe MAC are clock-gated and bypassed when PCIe interface is not enabled; enable monitoring of PCIe MAC memories by setting ICPU_CFG::PCIE_CFG.MEM_RING_CORE_ENA. This field must not be set when PCIe interface is not enabled.

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. The following flowchart shows how the detection logic sets flags and address.

FIGURE 3-31: 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.

The following table 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 3-42: MEMORIES WITH INTEGRITY SUPPORT

Index	Description
Others	For unlisted indexes, the recommended approach is to reboot the device.

The VCore-III CPU implements parity protection of all VCore-III cache structures as specified by MIPS architecture. For information about how to enable parity protection and interrupt on parity upsets, see the MIPS documentation.

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.

3.8.13 INTERRUPT CONTROLLER

This section provides information about the VCore-III interrupt controller.

The following table lists the registers associated with the interrupt controller.

TABLE 3-43: INTERRUPT CONTROLLER REGISTERS

Register	Description
ICPU_CFG::INTR_RAW	Current value of interrupt inputs
ICPU_CFG::INTR_BYPASS	Force non-sticky function
ICPU_CFG::INTR_TRIGGER	Configure edge or level sensitive events
ICPU_CFG::INTR_FORCE	Force events (for software debug)
ICPU_CFG::INTR_STICKY	Currently logged events
ICPU_CFG::INTR_ENA	Enable of interrupt sources
ICPU_CFG::INTR_ENA_CLR	Atomic clear of bits in INTR_ENA
ICPU_CFG::INTR_ENA_SET	Atomic set of bits in INTR_ENA
ICPU_CFG::INTR_IDENT	Currently enabled and interrupting sources
ICPU_CFG::DST_INTR_MAP	Mapping of interrupt sources to destinations
ICPU_CFG::DST_INTR_IDENT	Currently enabled, mapped, and interrupting sources per destination
ICPU_CFG::DEV_INTR_POL	Polarity of module interrupt inputs
ICPU_CFG::DEV_INTR_RAW	Current value of module interrupts
ICPU_CFG::DEV_INTR_BYPASS	Force non-sticky function for module interrupts
ICPU_CFG::DEV_INTR_TRIGGER	Configure edge or level sensitive events for module interrupts
ICPU_CFG::DEV_INTR_STICKY	Currently logged module interrupt events
ICPU_CFG::DEV_INTR_ENA	Enable of module interrupts
ICPU_CFG::DEV_INTR_IDENT	Currently interrupting and enabled module interrupts
ICPU_CFG::EXT_SRC_INTR_POL	Polarity of external interrupt inputs.
ICPU_CFG::EXT_DST_INTR_POL	Polarity of external interrupt outputs.
ICPU_CFG::EXT_DST_INTR_DRV	Drive mode for external interrupt outputs

The interrupt controller maps interrupt sources from VCore-III and switch core blocks, port modules, and external interrupt inputs to four interrupt destinations. Two interrupt destinations are mapped to the VCore-III CPU, and two can be transmitted from the device using the overlaid functions on GPIOs or using PCIe inband interrupt signaling.

The following table lists the available interrupt sources in the device.

TABLE 3-44: INTERRUPT SOURCES

Source Name	Description
DEV	Aggregated port module interrupt. This interrupt is asserted if there is an active and enabled interrupt from any of the device's port modules. See Port Module Interrupts. This interrupt has bit index 0 in INTR_* and DST_INTR_* registers.
EXT_SRC0	External interrupt source 0. See External Interrupts. This interrupt has bit index 1 in INTR_* and DST_INTR_* registers.
EXT_SRC1	External interrupt source 1. See External Interrupts. This interrupt has bit index 2 in INTR_* and DST_INTR_* registers.
TIMER0	Timer 0 interrupt. See Timers. This interrupt has bit index 3 in INTR_* and DST_INTR_* registers.
TIMER1	Timer 1 interrupt. See Timers. This interrupt has bit index 4 in INTR_* and DST_INTR_* registers.
TIMER2	Timer 2 interrupt. See Timers. This interrupt has bit index 5 in INTR_* and DST_INTR_* registers.
UART	UART interrupt. See UART Interrupt. This interrupt has bit index 6 in INTR_* and DST_INTR_* registers.

TABLE 3-44: INTERRUPT SOURCES (CONTINUED)

Source Name	Description
UART2	UART2 interrupt. See UART Interrupt. This interrupt has bit index 7 in INTR_* and DST_INTR_* registers.
TWI	TWI interrupt. See Two-Wire Serial Interface Interrupt. This interrupt has bit index 8 in INTR_* and DST_INTR_* registers.
SIMC	Serial Master Controller interrupt. See Serial Master Controller Interrupt. This interrupt has bit index 9 in INTR_* and DST_INTR_* registers.
SW0	Software interrupt 0. See Mailbox and Semaphores. This interrupt has bit index 10 in INTR_* and DST_INTR_* registers.
SW1	Software interrupt 1. See Mailbox and Semaphores. This interrupt has bit index 11 in INTR_* and DST_INTR_* registers.
SGPIO0	Serial GPIO interrupt 0. See SIO Interrupt. This interrupt has bit index 12 in INTR_* and DST_INTR_* registers.
GPIO	Parallel GPIO interrupt. See GPIO Interrupt. This interrupt has bit index 13 in INTR_* and DST_INTR_* registers.
MIIM0	MIIM Controller 0 interrupt. See MII Management Interrupt. This interrupt has bit index 14 in INTR_* and DST_INTR_* registers.
MIIM1	MIIM Controller 1 interrupt. See MII Management Interrupt. This interrupt has bit index 15 in INTR_* and DST_INTR_* registers.
FDMA	Frame DMA interrupt, see FDMA Events and Interrupts. This interrupt has bit index 16 in INTR_* and DST_INTR_* registers.
ANA	Analyzer interrupt. See Interrupt Handling. This interrupt has bit index 17 in INTR_* and DST_INTR_* registers.
PTP_RDY	Time stamp ready interrupt. See Hardware Time Stamping Module. This interrupt has bit index 18 in INTR_* and DST_INTR_* registers.
PTP_SYNC	PTP synchronization interrupt. See Master Timer. This interrupt has bit index 19 in INTR_* and DST_INTR_* registers.
ITGR	Memory integrity interrupt. See Section 3.8.12, Memory Integrity Monitor. This interrupt has bit index 20 in INTR_* and DST_INTR_* registers.
XTR_RDY	Extraction data ready interrupt. See Frame Extraction. This interrupt has bit index 21 in INTR_* and DST_INTR_* registers.
INJ_RDY	Injection ready interrupt. See Frame Injection. This interrupt has bit index 22 in INTR_* and DST_INTR_* registers.
PCIE	PCIe interrupt. See Power Management. This interrupt has bit index 23 in INTR_* and DST_INTR_* registers.

The following table lists the available interrupt destinations in the device.

TABLE 3-45: INTERRUPT DESTINATIONS

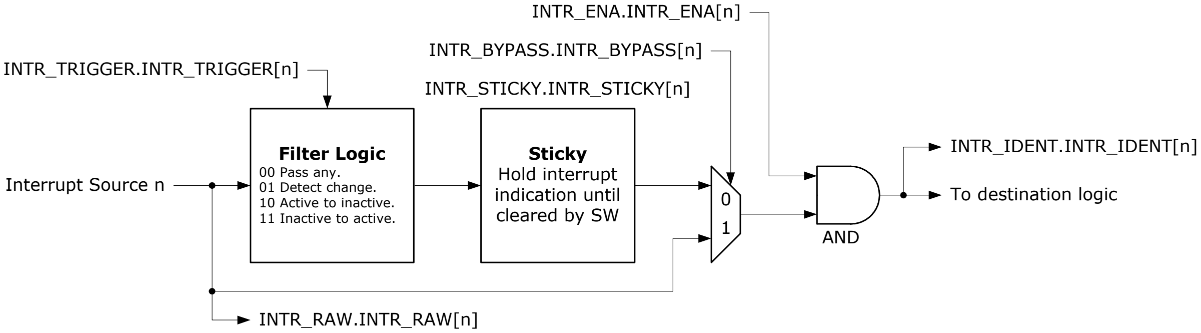
Destination Name	Description
CPU0	Interrupt 0 to VCore-III CPU. This interrupt has replication index 0 in DST_INTR_* registers.
CPU1	Interrupt 1 to VCore-III CPU. This interrupt has replication index 1 in DST_INTR_* registers.
EXT_DST0	External interrupt destination 0. See External Interrupts. This interrupt has replication index 2 in DST_INTR_* registers.
EXT_DST1	External interrupt destination 1. See External Interrupts. This interrupt has replication index 3 in DST_INTR_* registers.

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.

3.8.13.1 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. The following illustration shows the logic associated with a single interrupt source.

FIGURE 3-32: 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 two 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. For more information, see Interrupt Destination Configuration. An active and enabled interrupt source sets INTR_IDENT.INTR_IDENT[*n*].

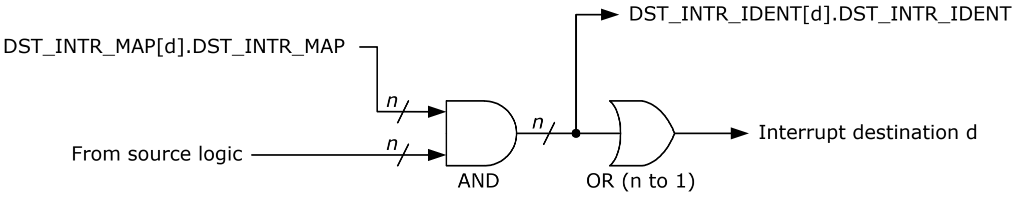
On rare occasions it is desirable to bypass the stickiness of interrupt sources and use INTR_RAW.INTR_RAW[*n*] directly instead of INTR_STICKY.INTR_STICKY[*n*]. Set INTR_BYPASS.INTR_BYPASS[*n*] to enable bypass and ignore INTR_STICKY and INTR_TRIGGER configurations.

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.

3.8.13.2 Interrupt Destination Configuration

The four 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. The following illustration shows the logic associated with a single interrupt destination.

FIGURE 3-33: INTERRUPT DESTINATION LOGIC


DST_INTR_IDENT[d].DST_INTR_IDENT
DST_INTR_MAP[d].DST_INTR_MAP
n
n
n
From source logic
AND
OR (n to 1)
Interrupt destination d

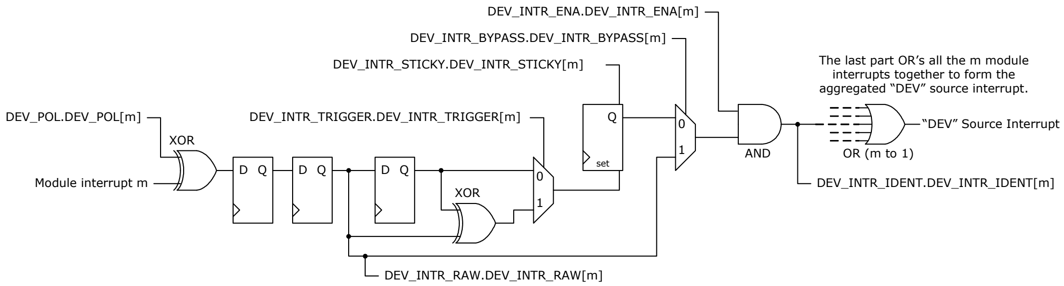
The interrupt destination can enable individual sources for interrupt by writing a mask to DST_INTR_MAP[d].DST_INTR_MAP. When a source is enabled in DST_INTR_MAP then an interrupt from this source will be propagated to the interrupt destination.

The currently active and enabled source interrupts for a destination can be seen by reading DST_INTR_IDENT[d].DST_INTR_IDENT.

3.8.13.3 Port Module Interrupts

Each port module can generate an interrupt. Because there are too 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 3-34: PORT MODULE INTERRUPT LOGIC


DEV_INTR_ENA.DEV_INTR_ENA[m]
DEV_INTR_BYPASS.DEV_INTR_BYPASS[m]
DEV_INTR_STICKY.DEV_INTR_STICKY[m]
The last part OR's all the m module interrupts together to form the aggregated "DEV" source interrupt.
DEV_POL.DEV_POL[m]
DEV_INTR_TRIGGER.DEV_INTR_TRIGGER[m]
XOR
Q
set
0
1
"DEV" Source Interrupt
AND
OR (m to 1)
Module interrupt m
D Q
D Q
D Q
XOR
0
1
DEV_INTR_IDENT.DEV_INTR_IDENT[m]
DEV_INTR_RAW.DEV_INTR_RAW[m]

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, see Interrupt Source Configuration for more information.

The final step when handling module interrupts is an aggregation of all individual module interrupts to the DEV source interrupt.

3.8.13.4 External Interrupts

The interrupt controller supports two external source interrupts (inputs to the device) and two external destination interrupts (outputs from the device). The external interrupts are mapped to GPIOs using overlaid functions. For more information about overlaid functions on the GPIOs for these signals, see Section 3.8.8.1, GPIO Overlaid Functions.

Source and destination interrupts works independently from each other and 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.

TABLE 3-46: EXTERNAL INTERRUPT PINS

Register	I/O	Description
IRQ0_IN/GPIO	I	External Source Interrupt 0. Polarity is configured in EXT_SRC_INTR_POL.EXT_INTR_POL[0].
IRQ1_IN/GPIO	I	External Source Interrupt 1. Polarity is configured in EXT_SRC_INTR_POL.EXT_INTR_POL[1].
IRQ0_OUT/GPIO	O	External Destination Interrupt 0. Polarity is configured in EXT_DST_INTR_POL.EXT_INTR_POL[0]. This interrupt can also be mapped to GPIO (replaces source interrupt).
IRQ1_OUT/GPIO	O	External Destination Interrupt 1. Polarity is configured in EXT_DST_INTR_POL.EXT_INTR_POL[1]. This interrupt can also be mapped to GPIO (replaces source interrupt).

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 required when multiple interrupt sources are hooked up to the same interrupt wire on the PCB and the wire is be pulled high with a resistor. Each interrupt source can then drive the wire low through open-collector output when they want to signal interrupt.

4.0 FEATURES

This section provides information about specific features supported by individual blocks in the device and how these features are administrated by configurations across the entire device. Examples of various standard features are included such as the support for different spanning tree versions and VLAN operations, as well as more advanced features such as QoS and VCAP.

4.1 Switch Control

This section provides information about the minimum requirements for switch operation.

4.1.1 SWITCH INITIALIZATION

The following initialization sequence is required to ensure proper operation of the switch.

1. Soft-reset the switch core:

   DEVCPU_GCB::SOFT_RST.SOFT_SWC_RST = 1
2. Wait for DEVCPU_GCB::SOFT_RST.SOFT_SWC_RST to clear.

   Initialize
   SYS::RESET_CFG.MEM_ENA = 1
   SYS::RESET_CFG.MEM_INIT = 1
3. Wait 100 µs for memories to initialize (SYS::RESET_CFG.MEM_INIT cleared).
4. Enable the switch core:

   SYS::RESET_CFG.CORE_ENA = 1
5. Enable each port module through QSYS:PORT:SWITCH_PORT_MODE.PORT_ENA = 1

4.2 Port Module Control

This section provides information about the features and configurations for port reset and port counters.

4.2.1 PORT RESET PROCEDURE

When changing a switch port’s mode of operation or restarting a switch port, the following port reset procedure must be followed:

1. Disable the MAC frame reception in the switch port.

   DEV::MAC_ENA_CFG.RX_ENA = 0
2. Disable traffic being sent to or from the switch port.

   QSYS:PORT:SWITCH_PORT_MODE_ENA = 0
3. Disable shaping to speed up flushing of frames.

   QSYS::PORT_MODE.DEQUEUE_DIS = 1
4. Wait at least the time it takes to receive a frame of maximum length at the port.

   Worst-case delays for 10 kilobyte jumbo frames are:
   - 8 ms on a 10M port
   - 800 µs on a 100M port
   - 80 µs on a 1G port
   - 32 µs on a 2.5G port
5. Disable HDX backpressure.

   SYS::FRONT_PORT_MODE.HDX_MODE = 0
6. Flush the queues associated with the port.

   REW:PORT:PORT_CFG.FLUSH_ENA = 1
7. Enable dequeuing from the egress queues.

   QSYS::PORT_MODE.DEQUEUE_DIS = 0
8. Wait until flushing is complete.

QSYS:PORT:SW_STATUS.EQ_AVAIL must return 0

9. Reset the switch port by setting the following reset bits in CLOCK_CFG:
   - DEV::CLOCK_CFG.MAC_TX_RST = 1
   - DEV::CLOCK_CFG.MAC_RX_RST = 1
   - DEV::CLOCK_CFG.PORT_RST = 1
10. Clear flushing again.

    REW:PORT:PORT_CFG.FLUSH_ENA = 0
11. Set up the switch port to the new mode of operation. Keep the reset bits in CLOCK_CFG set.
12. Release the switch port from reset by clearing the reset bits in CLOCK_CFG.

Note: It is not necessary to reset the SerDes macros.

4.2.2 PORT COUNTERS

The statistics collected in each port module provide monitoring of various events. This section describes how industry-standard Management Information Bases (MIBs) can be implemented using the counter set in the device. The following MIBs are considered.

- RMON statistics group (RFC 2819)
- IEEE 802.3-2005 Annex 30A counters
- SNMP interfaces group (RFC 2863)
- SNMP Ethernet-like group (RFC 3536)

4.2.2.1 RMON Statistics Group (RFC 2819)

The following table provides the mapping of RMON counters to port counters.

TABLE 4-1: MAPPING OF RMON COUNTERS TO PORT COUNTERS

RMON Counter	RX/TX	Switch Core Implementation
EtherStatsDropEvents	RX	C_RX_CAT_DROP + C_DR_TAIL + sum of C_DR_YELLOW_PRIO_x + sum of C_DR_GREEN_PRIO_x, where x is 0 through 7.
EtherStatsOctets	RX	C_RX_OCT
EtherStatsPkts	RX	C_RX_SHORT + C_RX_FRAG + C_RX_JABBER + C_RX_LONG + C_RX_SZ_64 + C_RX_SZ_65_127 + C_RX_SZ_128_255 + C_RX_SZ_256_511 + C_RX_SZ_512_1023 + C_RX_SZ_1024_1526 + C_RX_SZ_JUMBO
EtherStatsBroadcastPkts	RX	C_RX_BC
EtherStatsMulticastPkts	RX	C_RX_MC
EtherStatsCRCAlignErrors	RX	C_RX_CRC
EtherStatsUndersizePkts	RX	C_RX_SHORT
EtherStatsOversizePkts	RX	C_RX_LONG
EtherStatsFragments	RX	C_RX_FRAG
EtherStatsJabbers	RX	C_RX_JABBER
EtherStatsPkts64Octets	RX	C_RX_SZ_64
EtherStatsPkts65to127Octets	RX	C_RX_SZ_65_127
EtherStatsPkts128to255Octets	RX	C_RX_SZ_128_255
EtherStatsPkts256to511Octets	RX	C_RX_SZ_256_511
EtherStatsPkts512to1023Octets	RX	C_RX_SZ_512_1023
EtherStatsPkts1024to1518Octets	RX	C_RX_SZ_1024_1526
EtherStatsDropEvents	TX	C_TX_DROP + C_TX_AGE
EtherStatsOctets	TX	C_TX_OCT

TABLE 4-1: MAPPING OF RMON COUNTERS TO PORT COUNTERS (CONTINUED)

RMON Counter	RX/TX	Switch Core Implementation
EtherStatsPkts	TX	C_TX_SZ_64 + C_TX_SZ_65_127 + C_TX_SZ_128_255 + C_TX_SZ_256_511 + C_TX_SZ_512_1023 + C_TX_SZ_1024_1526 + C_TX_SZ_JUMBO
EtherStatsBroadcastPkts	TX	C_TX_BC
EtherStatsMulticastPkts	TX	C_TX_MC
EtherStatsCollisions	TX	C_TX_COL
EtherStatsPkts64Octets	TX	C_TX_SZ_64
EtherStatsPkts65to127Octets	TX	C_TX_SZ_65_127
EtherStatsPkts128to255Octets	TX	C_TX_SZ_128_255
EtherStatsPkts256to511Octets	TX	C_TX_SZ_256_511
EtherStatsPkts512to1023Octets	TX	C_TX_SZ_512_1023
EtherStatsPkts1024to1518Octets	TX	C_TX_SZ_1024_1526

4.2.2.2 IEEE 802.3-2005 Annex 30A Counters

This section provides the mapping of IEEE 802.3-2005 Annex 30A counters to port counters. Only counter groups with supported counters are listed.

TABLE 4-2: MANDATORY COUNTERS

Counter	Rx/Tx	Switch Core Implementation
aFramesTransmittedOK	TX	C_TX_SZ_64 + C_TX_SZ_65_127 + C_TX_SZ_128_255 + C_TX_SZ_256_511 + C_TX_SZ_512_1023 + C_TX_SZ_1024_1526 + C_TX_SZ_JUMBO
aSingleCollisionFrames	TX	Not available.
aMultipleCollisionFrames	TX	Not available.
aFramesReceivedOK	RX	Sum of C_RX_GREEN_PRIO_x + C_RX_YELLOW_PRIO_x, where x is 0 through 7.
aFrameCheckSequenceErrors	RX	Not available. C_RX_CRC is the sum of FCS and alignment errors.
aAlignmentErrors	RX	Not available. C_RX_CRC is the sum of FCS and alignment errors.

TABLE 4-3: OPTIONAL COUNTERS

Counter	RX/TX	Switch Core Implementation
aMulticastFramesXmittedOK	TX	C_TX_MC
aBroadcastFramesXmittedOK	TX	C_TX_BC
aMulticastFramesReceivedOK	RX	C_RX_MC
aBroadcastFramesReceivedOK	RX	C_RX_BC
aInRangeLengthErrors	RX	Not available
aOutOfRangeLengthField	RX	Not available
aFrameTooLongErrors	RX	C_RX_LONG

TABLE 4-4: RECOMMENDED MAC CONTROL COUNTERS

Counter	RX/TX	Switch Core Implementation
aMACControlFramesTransmitted	TX	Not available
aMACControlFramesReceived	RX	C_RX_CONTROL

TABLE 4-4: RECOMMENDED MAC CONTROL COUNTERS (CONTINUED)

Counter	RX/TX	Switch Core Implementation
aUnsupportedOpcodesReceived	RX	Not available

TABLE 4-5: PAUSE MAC CONTROL RECOMMENDED COUNTERS

Counter	RX/TX	Switch Core Implementation
aPauseMACControlFramesTransmitted	TX	C_TX_PAUSE
aPauseMACControlFramesReceived	RX	C_RX_PAUSE

4.2.2.3 SNMP Interfaces Group (RFC 2863)

The following table provides the mapping of SNMP interfaces group counters to port counters.

TABLE 4-6: MAPPING OF SNMP INTERFACES GROUP COUNTERS TO PORT COUNTERS

Counter	RX/TX	Switch Core Implementation
IfInOctets	RX	C_RX_OCT
IfInUcastPkts	RX	C_RX_UC
IfInNUcastPkts	RX	C_RX_BC + C_RX_MC
IfInBroadcast (RFC 1573)	RX	C_RX_BC
IfInMulticast (RFC 1573)	RX	C_RX_MC
IfInDiscards	RX	C_DR_TAIL + C_RX_CAT_DROP
IfInErrors	RX	C_RX_CRC + C_RX_SHORT + C_RX_FRAG + C_RX_JABBER + C_RX_LONG
IfInUnknownProtos	RX	Always zero.
IfOutOctets	TX	C_TX_OCT
IfOutUcastPkts	TX	C_TX_UC
IfOutNUcastPkts	TX	C_TX_BC + C_TX_MC
ifOutMulticast (RFC 1573)	TX	C_TX_MC
ifOutBroadcast (RFC 1573)	TX	C_TX_BC
IfOutDiscards	TX	Always zero.
IfOutErrors	TX	C_TX_DROP + C_TX_AGE

4.2.2.4 SNMP Ethernet-Like Group (RFC 3536)

The following table provides the mapping of SNMP Ethernet-like group counters to port counters.

TABLE 4-7: MAPPING OF SNMP ETHERNET-LIKE GROUP COUNTERS TO PORT COUNTERS

Counter	RX/TX	Switch Core Implementation
dot3StatsAlignmentErrors	RX	Not available. C_RX_CRC is the sum of FCS and alignment errors.
dot3StatsFCSErrors	RX	Not available. C_RX_CRC is the sum of FCS and alignment errors.
dot3StatsSingleCollisionFrames	TX	Not available.
dot3StatsMultipleCollisionFrames	TX	Not available.
dot3StatsSQETestErrors	RX	Not applicable.
dot3StatsDeferredTransmissions	TX	Not available.
dot3StatsLateCollisions	TX	Not available. C_TX_DROP is the sum of Late collisions and Excessive collisions.
dot3StatsExcessiveCollisions	TX	Not available. C_TX_DROP is the sum of Late collisions and Excessive collisions.

TABLE 4-7: MAPPING OF SNMP ETHERNET-LIKE GROUP COUNTERS TO PORT COUNTERS

Counter	RX/TX	Switch Core Implementation
dot3StatsInternalMacTransmitErrors	TX	Not applicable. Always 0.
dot3StatsCarrierSenseErrors	TX	Not available.
dot3StatsFrameTooLongs	RX	C_RX_LONG.
dot3StatsInternalMacReceiveErrors	RX	Not applicable. Always 0.
dot3InPauseFrames	RX	C_RX_PAUSE.
dot3OutPauseFrames	TX	C_TX_PAUSE.

4.3 Layer-2 Switch

This section describes the following Layer-2 switch features.

- Switching
- VLAN and GVRP
- Rapid and Multiple Spanning Tree
- Link aggregation
- Port-based access control
- Mirroring
- SNMP support

4.3.1 BASIC SWITCHING

Basic switching covers forwarding, address learning, and address aging.

4.3.1.1 Forwarding

The device contains a Layer-2 switch and frames are forwarded using Layer-2 information only. Exceptions to this are possible using VCAP capabilities. For example, to provide source-specific IP multicast forwarding.

The switch is designed to comply with the IEEE Bridging standard in Std 802.1D and the IEEE VLAN standard in Std 802.1Q:

- Unicast frames are forwarded to a single destination port that corresponds to the DMAC.
- Multicast frames are forwarded to multiple ports determined by the DMAC multicast group. The CPU configures multicast groups in the MAC table and the port group identifier (PGID) table. A multicast group can span across any set of ports.
- Broadcast frames (DMAC = FF-FF-FF-FF-FF-FF) are, by default, flooded to all ports except the ingress port. Also, in compliance with the standard, a unicast or multicast frame with unknown DMAC is flooded to all ports except the ingress port. It is possible to configure flood masks to restrict the flooding of frames. There are separate flood masks for the following frame types:
- Unicast (ANA::FLOODING.FLD_UNICAST)
- Layer 2 multicast (ANA::FLOODING.FLD_MULTICAST)
- Layer 2 broadcast (ANA::FLOODING.FLD_BROADCAST)
- IPv4 multicast data (ANA::FLOODING_IPMC.FLD_MC4_DATA)
- IPv4 multicast control (ANA::FLOODING_IPMC.FLD_MC4_CTRL)
- IPv6 multicast data (ANA::FLOODING_IPMC.FLD_MC6_DATA)
- IPv6 multicast control (ANA::FLOODING_IPMC.FLD_MC6_CTRL)

For frames with a known destination MAC address, the destination mask comes from an entry in the port group identifier table (ANA::PGID). The PGID table contains 92 entries (entry 0 through 91), where entry 0 through 63 are used for destination masks. The remaining PGID entries are used for other parts of the forwarding and are described below.

The following table shows the PGID table organization.

TABLE 4-8: PORT GROUP IDENTIFIER TABLE ORGANIZATION

Entry Type	Number
Unicast entries	0 – 11 (including CPU)

TABLE 4-8: PORT GROUP IDENTIFIER TABLE ORGANIZATION

Entry Type	Number
Multicast entries	12 – 63
Aggregation Masks	64 – 79
Source Masks	80 – 91

The unicast entries contains only the port number corresponding to the entry number.

Destination masks for multicast groups must be manually entered through the CPU into the destination masks table. IPv4 and IPv6 multicast entries can also be entered using direct encoding in the MAC table, where the destination masks table is not used. For information about forwarding and configuring destination masks, see <need reference>.

The aggregation masks ensures that a frame is forwarded to exactly one member of an aggregation group.

For all forwarding decisions, a source mask prevents frames from being sent back to the ingress port. The source mask removes the ingress port from the destination mask.

All ports are enabled for receiving frames by default. This can be disabled by clearing ANA:PORT:PORT_CFG.RECV_ENA.

4.3.1.2 Address Learning

The learning process minimizes the flooding of frames. A frame's source MAC address is learned together with its VID. Each entry in the MAC table is uniquely identified by a (MAC,VID) pair. In the forwarding process, a frame's (DMAC,VID) pair is used as the key for the MAC table lookup.

The learning of unknown SMAC addresses can be either hardware-based or CPU-based. The following list shows the available learn schemes, which can be configured per port:

- **Hardware-based learning** autonomously adds entries to the MAC table without interaction from the CPU. Use the following configuration:
- ANA:PORT:PORT_CFG.LEARN_ENA = 1
- ANA:PORT:PORT_CFG.LEARNCPU = 0
- ANA:PORT:PORT_CFG.LEARNDROP = 0
- ANA:PORT:PORT_CFG.LEARNAUTO = 1
- **CPU-based learning** copies frames with unknown SMACs, or when the SMAC appears on a different port, to the CPU. These frames are forwarded as usual. Use the following configuration.
- ANA:PORT:PORT_CFG.LEARN_ENA = 1
- ANA:PORT:PORT_CFG.LEARNCPU = 1
- ANA:PORT:PORT_CFG.LEARNDROP = 0
- ANA:PORT:PORT_CFG.LEARNAUTO = 0
- **Secure CPU-based learning** is similar to CPU-based learning, except that it allows the CPU to verify the SMAC addresses before both learning and forwarding. Secure CPU-based learning redirects frames with unknown SMACs, or when the SMAC appears on a different port, to the CPU. These frames are not forwarded by hardware. Use the following configuration.
- ANA::PORT_CFG.LEARN_ENA = 1
- ANA::PORT_CFG.LEARNCPU = 1
- ANA::PORT_CFG.LEARNDROP = 1
- ANA::PORT_CFG.LEARNAUTO = 0
- **No learning** where all learn frames are discarded. Frames with known SMAC in the MAC table are forwarded by hardware. Use the following configuration.
- ANA:PORT:PORT_CFG.LEARN_ENA = 1
- ANA:PORT:PORT_CFG.LEARNCPU = 0
- ANA:PORT:PORT_CFG.LEARNDROP = 1
- ANA:PORT:PORT_CFG.LEARNAUTO = 0

Frames forwarded to the CPU for learning can be extracted from the CPU extraction queue configured in ANA:PORT:CPUQ_CFG.CPUQ_LRN.

During CPU-based learning, the rate of frames subject to learning being copied or redirected to the CPU can be controlled with the learn storm policer (ANA::STORMLIMIT_CFG[3]). This policer puts a limit on the number of frames per second that are subject to learning being copied or redirected to the CPU. The learn frames storm policer can help prevent a CPU from being overloaded when performing CPU based learning.

4.3.1.3 MAC Table Address Aging

To keep the MAC table updated, an aging scan is conducted to remove entries that were not recently accessed. This ensures that stations that have moved to a new location are not permanently prevented from receiving frames in their new location. It also frees up MAC table entries occupied by obsolete stations to give room for new stations.

In IEEE 802.1D, the recommended period for aging-out entries in the MAC address table is 300 seconds per entry. The device aging implementation checks for the aging-out of all the entries in the table. The first age scan sets the age bit for every entry in the table. The second age scan removes entries where the age bit has not been cleared since the first age scan. An entry's age bit is cleared when a received frame's (SMAC, VID) matches an entry's (MAC, VID); that is, the station is active and transmits frames. To ensure that 300 seconds is the longest an entry can reside inaccessible (and unchanged) in the table, the maximum time between age scans is 150 seconds.

The device can conduct age scans in two ways:

- Automatic age scans
- CPU initiated age scans

When using automatic aging, the time between age scans is set in the ANA::AUTOAGE register in steps of 1 second, in the range from 1 second to 12 days.

When using CPU-initiated aging, the CPU implements the timing between age scans. A scan is initiated by sending an aging command to the MAC address table (ANA::MACACCESS.MAC_TABLE_CMD).

The CPU-controlled age scan process can conveniently be used to flush the entire MAC table by conducting two age scans, one immediately after the other.

Flushing selective MAC table entries is also possible. Incidents that require MAC table flushing are:

- Reconfiguration of Spanning Tree protocol port states, which may cause station moves to occur.
- If there is a link failure notification (identified by a PHY layer device), flush the MAC table on the specific port where the link failed.

To deal with these incidents, the age scan process is configurable to run only for entries learned on a specified port or for a specified VLAN (ANA::ANAGEFIL.VID_VAL). The filters can also be combined to do aging on entries that match both the specific port and the specific VLAN.

Single entries can be flushed from the MAC table by sending the FORGET command to the MAC address table.

4.3.2 STANDARD VLAN OPERATION

This section provides information about configuring and operating the device as a standard VLAN-aware switch. Subsequent sections discuss the switch as a Q-in-Q enabled provider bridge and the use of private VLANs and asymmetric VLANs.

The following table lists the port module registers for standard VLAN operation.

TABLE 4-9: PORT MODULE REGISTERS FOR STANDARD VLAN OPERATION

Register/Register Field	Description	Replication
MAC_TAGS_CFG	Allows tagged frames to be 4 bytes longer than the length configured in MAC_MAXLEN_CFG.	Per port

The following table lists the analyzer configurations and status bits for standard VLAN operation.

TABLE 4-10: ANALYZER REGISTERS FOR STANDARD VLAN OPERATION

Register/Register Field	Description	Replication
DROP_CFG.DROP_UNTAGGED_ENA	Discard untagged frames.	Per port
DROP_CFG.DROP_C_TAGGED_ENA	Discard VLAN tagged frames.	Per port
DROP_CFG.DROP_PRIO_C_ TAGGED_ENA	Discard priority tagged frames.	Per port

TABLE 4-10: ANALYZER REGISTERS FOR STANDARD VLAN OPERATION (CONTINUED)

Register/Register Field	Description	Replication
VLAN_CFG.VLAN_AWARE_ENA	Use incoming VLAN tags in VLAN classification.	Per port
VLAN_CFG.VLAN_POP_CNT	Remove VLAN tags from frames in the rewriter.	Per port
VLAN_CFG.VLAN_DEI VLAN_CFG.VLAN_PCP VLAN_CFG.VLAN_VID	Ingress port VLAN configuration.	Per port
VLANMASK	Per-port VLAN ingress filtering enable.	None
ANEVENTS.VLAN_DISCARD	A sticky bit indicating that a frame was dropped due to lack of VLAN membership of source port.	None
ADVLEARN.VLAN_CHK	Disable learning for frames discarded due to source port VLAN membership check.	None
VLANACCESS	VLAN table command. For indirect access to configuration of the 4096 VLANs.	None
VLANTIDX	VLAN table index. For indirect access to configuration of the 4096 VLANs.	None
AGENCTRL.FID_MASK	Enable shared VLAN learning.	None
CPU_FWD_GARP_CFG	Enable capture of frames with reserved GARP DMAC addresses, including GVRP for VLAN registration. Per-address configuration.	Per port
CPUQ_8021_CFG.CPUQ_GARP_VAL	CPU queue for captured GARP frames.	Per GARP address

The following table lists the rewriter registers for standard VLAN operation.

TABLE 4-11: REWRITER REGISTERS FOR STANDARD VLAN OPERATION

Register/Register Field	Description	Replication
TAG_CFG	Egress VLAN tagging configuration.	Per port
PORT_VLAN_CFG	Egress port VLAN configuration.	Per port

In a VLAN-aware switch, each port is a member of one or more virtual LANs. Each incoming frame must be assigned a VLAN membership and forwarded according to the assigned VID. The following information draws on the definitions and principles of operations in IEEE 802.1Q. Note that the switch supports more features than mentioned in the following section, which only describes the basic requirements for a VLAN aware switch.

Standard VLAN operation is configured individually per switch port using the following configuration:

- MAC_TAGS_CFG.VLAN_AWR_ENA = 1
  MAC_TAGS_CFG.VLAN_LEN_AWR_ENA = 1
- VLAN_CFG.VLAN_AWARE_ENA = 1,
  VLAN_CFG.VLAN_POP_CNT = 1

Each switch port has an Acceptable Frame Type parameter, which is set to Admit Only VLAN tagged frames or Admit All Frames:

- Admit Only VLAN-tagged frames:
- DROP_CFG.DROP_UNTAGGED_ENA = 1
- DROP_CFG.DROP_PRIO_C_TAGGED_ENA = 1
- DROP_CFG.DROP_C_TAGGED = 0
- Admit All Frames:
- DROP_CFG.DROP_UNTAGGED_ENA = 0
- DROP_CFG.DROP_PRIO_C_TAGGED_ENA = 0
- DROP_CFG.DROP_C_TAGGED = 0

Frames that are not discarded are subject to the VLAN classification. Untagged and priority-tagged frames are classified to a Port VLAN Identifier (PVID). The PVID is configured per port in `VLAN_CFG.VLAN_VID`. Tagged frames are classified to the VID given in the frame's tag. For more information about VLAN classification, see Section 2.7.4, VLAN Classification.

4.3.2.1 Forwarding

Forwarding is always based on the combination of the classified VID and the destination MAC address. By default, all switch ports are members of all VLANs. This can be changed in VLANACCESS and VLANTIDX where port masks per VLAN are set up.

4.3.2.2 Ingress Filtering

VLAN ingress filtering can be enabled per switch port with the register VLANMASK.

The filter checks for all incoming frames to determine if the ingress port is a member of the VLAN to which the frame is classified. If the port is not a member, the frame is discarded. Whenever a frame is discarded due to lack of VLAN membership, the ANEVENTS.VLAN_DISCARD sticky bit is set. To ensure that VLAN ingress filtered frames are not learned, ADVLEARN.VLAN_CHK must be set.

4.3.2.3 GVRP

GARP VLAN Registration Protocol (GVRP) is used to propagate VLAN configurations between bridges. On a GVRP-enabled switch, all GVRP frames must be redirected to the CPU for further processing. The GVRP frames use a reserved GARP MAC address (01-80-C2-00-00-21) and can be redirected to the CPU by setting bit 1 in the analyzer register CPU_FWD_GARP_CFG.

4.3.2.4 Shared VLAN Learning

The device can be configured for either Independent VLAN learning or Shared VLAN learning. Independent VLAN learning is the default.

Shared VLAN learning, where multiple VLANs map to the same filtering database, is enabled through Filter Identifiers (FIDs). Basically, this means that learning is unique for a (MAC, FID) set and that a learned MAC address is learned for all VIDs that map to the FID. Shared VLAN learning is configured in FID_MAP per VLAN. The device supports 64 different FIDs. Any number of VLANs can map to the same FID.

4.3.2.5 Untagging

An untagged set can be configured for each egress port, which defines the VIDs for which frames are transmitted untagged. The untagged set can consist of zero, one, or all VIDs. For all VIDs not in the untagged set, frames are transmitted tagged. The available configurations in the rewriter are:

- The untagged set is empty.
  REW:PORT:TAG_CFG.TAG_CFG = 3
- The untagged set consists of all VIDs.
  REW:PORT:TAG_CFG.TAG_CFG = 0
- The untagged set consists of one VID <VID>.
  REW:PORT:TAG_CFG.TAG_CFG = 1
  REW:PORT:PORT_VLAN_CFG.PORT_VID = <VID>

Optionally, frames received as priority-tagged frames (VID = 0) can also be transmitted as untagged (REW:PORT:TAG_CFG.TAG_CFG=2).

Port-Based VLAN Example

Situation:

Ports 0 and 1 are isolated from ports 2 and 3 using port-based VLANs. Ports 0 and 1 are assigned port VID 1 and ports 2 and 3 port VID 2. All frames in the network are untagged.

Resolution:

```
# Port module configuration of ports 0 - 1.
# Configure the ports to always use the port VID.
VLAN_CFG.VLAN_AWARE_ENA = 0
```

```
# Allow only untagged frames.
DROP_CFG.DROP_UNTAGGED_ENA = 0
DROP_CFG.DROP_PRIO_C_TAGGED = 1
DROP_CFG.DROP_C_TAGGED = 1
# Configure the port VID to 1.
VLAN_CFG.VLAN_VID = 1

# Port module configuration of ports 2 – 3.
# Same as for ports 0-1, except that the port VID is set to 2.
VLAN_CFG.VLAN_VID = 2
# Analyzer configuration.
# Configure VLAN 1 to contain ports 0-1.
VLANTIDX.INDEX = 1
VLANTIDX.VLAN_PRIV_VLAN = 0
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0
VLANTIDX.VLAN_SRC_CHK = 1
VLANACCESS.VLAN_PORT_MASK = 0x03
VLANACCESS.VLAN_TBL_CMD = 2
# Configure VLAN 2 to contain ports 2-3.
VLANTIDX.INDEX = 2
VLANTIDX.VLAN_PRIV_VLAN = 0
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0
VLANTIDX.VLAN_SRC_CHK = 1
VLANACCESS.VLAN_PORT_MASK = 0x0C
VLANACCESS.VLAN_TBL_CMD = 2
```

4.3.3 PROVIDER BRIDGES AND Q-IN-Q OPERATION

The following table lists the port module configurations for provider bridge VLAN operation.

TABLE 4-12: PORT MODULE CONFIGURATIONS FOR PROVIDER BRIDGE VLAN OPERATION

Register/Register Field	Description	Replication
MAC_TAGS_CFG	Allow single tagged frames to be 4 bytes longer and double-tagged frames to be 8 bytes longer than the length configured in MAC_MAXLEN_CFG.	Per port

The following table lists the port module configurations for provider bridge VLAN operation.

TABLE 4-13: SYSTEM CONFIGURATIONS FOR PROVIDER BRIDGE VLAN OPERATION

Register/Register Field	Description	Replication
VLAN_E-TYPE_CFG.VLAN_S_TAG_E-TYPE_VAL	TPID for S-tagged frames. EtherType 0x88A8 and the configurable value VLAN_E-TYPE_CFG.VLAN_S_TAG_ETYPE_VAL are identified as the S-tag identifier.	Per port

The following table lists the analyzer configurations for provider bridge VLAN operation.

TABLE 4-14: ANALYZER CONFIGURATIONS FOR PROVIDER BRIDGE VLAN OPERATION

Register/Register Field	Description	Replication
DROP_CFG.DROP_UNTAGGED_ENA	Discard untagged frames.	Per port
DROP_CFG.DROP_S_TAGGED_ENA	Discard VLAN S-tagged frames.	Per port
DROP_CFG.DROP_PRIO_S_TAGGED_ENA	Discard priority S-tagged frames.	Per port
VLAN_CFG.VLAN_AWARE_ENA	Use incoming VLAN tags in VLAN classification.	Per port
VLAN_CFG.VLAN_POP_CNT	Remove VLAN tags from frames in the rewriter.	Per port

TABLE 4-14: ANALYZER CONFIGURATIONS FOR PROVIDER BRIDGE VLAN OPERATION

Register/Register Field	Description	Replication
VLAN_CFG.VLAN_TAG_TYPE	Tag type for untagged frames (Customer tag or service tag).	Per port
VLAN_CFG.VLAN_INNER_TAG_ENA	Use inner tag for VLAN classification instead of outer tag.	Per port
VLAN_CFG.VLAN_DEI VLAN_CFG.VLAN_PCP VLAN_CFG.VLAN_VID	Ingress port VLAN configuration.	Per port
VLANACCESS	VLAN table command. For indirect access to configuration of the 4096 VLANs.	None
VLANTIDX	VLAN table index. For indirect access to configuration of the 4096 VLANs.	None

The device supports the standard provider bridge features in IEEE 802.1ad (Provider Bridges). Features related to provider bridges are:

- Support for multiple tag headers (EtherTypes 0x8100, 0x88A8, and a programmable value are recognized as tag header EtherTypes)
- Pushing and popping of up to two VLAN tags
- Selective VLAN classification using either inner or outer VLAN tag
- Translating VLAN tag headers at ingress and/or at egress (using the IS1 and ES0 TCAMs)
- Enabling or disabling learning per VLAN

The following section discusses briefly how to configure these different features in the switch.

The device supports multiple VLAN tags. They can be used in MAN applications as a provider bridge, aggregating traffic from numerous independent customer LANs into the MAN space. One of the purposes of the provider bridge is to recognize and use VLAN tags so that the VLANs in the MAN space can be used independent of the customers' VLANs. This is accomplished by adding a VLAN tag with a MAN-related VID for frames entering the MAN. When leaving the MAN, the tag is stripped, and the original VLAN tag with the customer-related VID is again available. This provides a tunneling mechanism to connect remote customer VLANs through a common MAN space without interfering with the VLAN tags. All tags use EtherType 0x8100 for customer tags and EtherType 0x88A8, or a programmable value, for service provider tags.

In cases where a given service VLAN only has two member ports on the switch, the learning can be disabled for the particular VLAN (VLANTIDX.VLAN_LEARN_DISABLE) and can rely on flooding as the forwarding mechanism between the two ports. This way, the MAC table requirements are reduced.

4.3.3.0.1 MAN Access Switch Example

Situation:

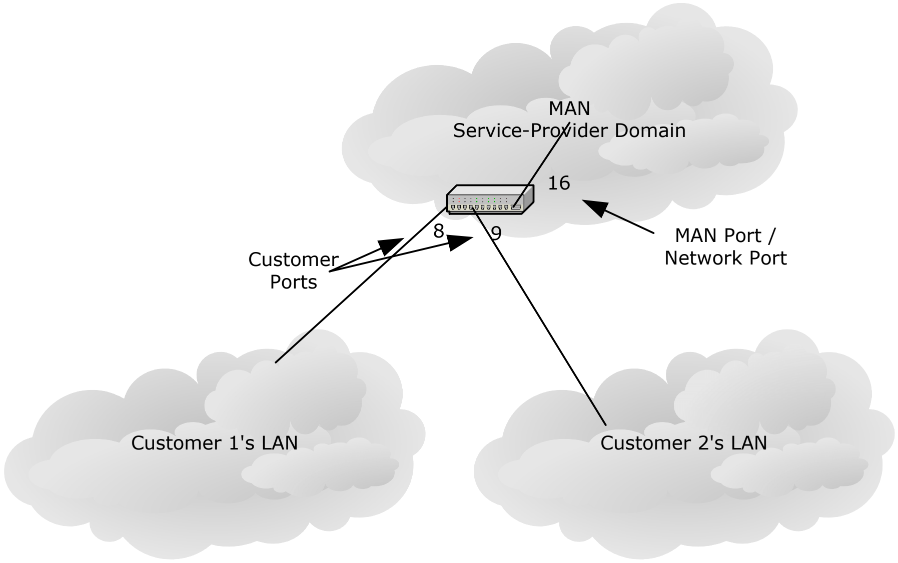
The following is an example of setting up the device as a MAN access switch with the following requirements:

- Customer ports are aggregated into a network port for tunneling through the MAN to access remote VLANs.
- Local switching between ports of the different customers must be eliminated.
- Frames must be label-switched from network port to correct customer port without need for MAC address learning.

FIGURE 4-1: MAN ACCESS SWITCH SETUP

Frames in This Segment

Service Provider Tag (Outer Tag)		Customer Tag (Inner Tag)		
EtherType	VID	EtherType	VID	Description
0x88A8	1	0x8100	1	Frames to/from customer 1's VLAN 1
0x88A8	1	0x8100	118	Frames to/from customer 1's VLAN 118
0x88A8	1	0x8100	0	Priority-tagged frames to/from customer 1
0x88A8	2	0x8100	1	Frames to/from customer 2's VLAN 1
0x88A8	2	0x8100	4	Frames to/from customer 2's VLAN 4
0x88A8	2	N/A	N/A	Untagged frames to/from customer 2



Frames in This Segment

Customer Tag		
EtherType	VID	Description
0x8100	1	Frames in customer 1's VLAN 1
0x8100	118	Frames in customer 1's VLAN 118
0x8100	0	Customer 1's priority-tagged frames

Frames in This Segment

Customer Tag		
EtherType	VID	Description
0x8100	1	Frames in customer 2's VLAN 1
0x8100	4	Frames in customer 2's VLAN 4
N/A	N/A	Customer 2's untagged frames

This example is typically accomplished by letting each customer port have a unique port VID (PVID), which is used in the outer VLAN tag (the service provider tag). In the MAN, the VID directly indicates the customer port from which the frame is received or the customer port to which the frame is going.

A customer port is VLAN-unaware and classifies to a port-based VLAN. In the egress direction of the customer port, frames are transmitted untagged, which facilitates the stripping of the outer tag. That is, the provider tag is stripped, but the customer tag is kept. The port must allow frames with a maximum size of 1522 bytes.

Resolution:

```
# Configuration of customer 1’s port (port 8).
# Allow for a single VLAN tag in the length check and set the maximum length without VLAN
# tag to 1518 bytes.
MAC_TAGS_CFG.VLAN_LEN_AWR_ENA = 1
MAC_TAGS_CFG.VLAN_AWAR_ENA = 1
MAC_MAXLEN_CFG.MAX_LEN = 1518
```

```
# Configure the port to leave any incoming tags in the frame and to ignore any
# incoming VLAN tags in the VLAN classification. The port VID is always used in the
# VLAN classification.
VLAN_CFG.VLAN_POP_CNT = 0
VLAN_CFG.VLAN_AWARE_ENA = 0
# Allow both C-tagged and untagged frames coming in to the device to also support customer traffic
not using VLANs to be carried across the MAN.
DROP_CFG.DROP_UNTAGGED_ENA = 0
DROP_CFG.DROP_C_TAGGED = 0
DROP_CFG.DROP_PRIO_C_TAGGED = 0
DROP_CFG.DROP_S_TAGGED = 1
DROP_CFG.DROP_PRIO_S_TAGGED = 1
# Use service provider tagging when frames from this port exit the switch.
# (EtherType 0x88A8).
VLAN_CFG.VLANTAG_TYPE = 1
# Configure the port VID to 1.
VLAN_CFG.VLAN_VID = 1
# Configure the egress side of the port to not insert tags.
# (The service provider tags are stripped in the ingress side of the MAN port).
TAG_CFG.TAG_CFG = 0
# Configuration of customer 2's port (port 9).
# Same as for customer 1’s port (port 8), except that the port VID is set to 2.
VLAN_CFG.VLAN_VID = 2
# Configuration of the network port (port 16).
# MAN traffic in transit between network ports is supported by configuring all network
# ports as follows:
# Allow for two VLAN tags in the length check and set the max length without
# VLAN tags to 1518 bytes.
MAC_TAGS_CFG.VLAN_LEN_AWR_ENA = 1
MAC_TAGS_CFG.VLAN_AWAR_ENA = 1
MAC_TAGS_CFG.PB_ENA =1
MAC_MAXLEN_CFG.MAX_LEN = 1518
# Configure the port to use incoming VLAN tags in the VLAN classification,
# and to remove the first (outer) VLAN tag (the service tag) from incoming frames.
VLAN_CFG.VLAN_POP_CNT = 1
VLAN_CFG.VLAN_AWARE_ENA = 1
# Allow only S-tagged frames.
DROP_CFG.DROP_UNTAGGED_ENA = 1
DROP_CFG.DROP_C_TAGGED = 1
DROP_CFG.DROP_PRIO_C_TAGGED = 1
DROP_CFG.DROP_S_TAGGED = 0
DROP_CFG.DROP_PRIO_S_TAGGED = 0
# The tag type is unused on the network port
VLAN_CFG.VLANTAG_TYPE = 0
# Configure the egress side of the port to insert tags.
TAG_CFG.TAG_CFG = 1
# Common configuration in the analyzer.
# Configure VLAN 1 to contain customer 1’s port (port 8) and the network port
# (port 16). Disable learning in VLAN 1. Ingress filtering is don’t care for port
# based VLANs.
VLANTIDX.INDEX = 1
VLANTIDX.VLAN_PRIV_VLAN = 0 (don’t care, for this example)
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 1
VLANTIDX.VLAN_SRC_CHK = 0 (don’t care, for this example)
VLANACCESS.VLAN_PORT_MASK = 0x00010100
VLANACCESS.VLAN_TBL_CMD = 2
# Configure VLAN 2 to contain customer 2's port (port 9) and the network port
# (port 16). Disable learning in VLAN 2. Ingress filtering is don’t-care for port
# based VLANs.
VLANTIDX.INDEX = 2
VLANTIDX.VLAN_PRIV_VLAN = 0 (don’t care, for this example)
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 1
VLANTIDX.VLAN_SRC_CHK = 0 (don’t care, for this example)
VLANACCESS.VLAN_PORT_MASK = 0x00010200
VLANACCESS.VLAN_TBL_CMD = 2
```

4.3.4 PRIVATE VLANS

The following table lists the analyzer configuration registers for private VLAN support.

TABLE 4-15: PRIVATE VLAN CONFIGURATION REGISTERS

Register	Description	Replication
VLANACCESS	VLAN table command. For indirect access to configuration of the 4096 VLANs.	None
VLANTIDX	VLAN table index. For indirect access to configuration of the 4096 VLANs.	None
ISOLATED_PORTS	VLAN port mask indicating isolated ports in private VLANs.	None
COMMUNITY_PORTS	VLAN port mask indicating community ports in private VLANs.	None

When a VLAN is configured to be a private VLAN, communication between ports within that VLAN can be prevented. Two application examples are:

- Customers connected to an ISP can be members of the same VLAN, but they are not allowed to communicate with each other within that VLAN.
- Servers in a farm of web servers in a Demilitarized Zone (DMZ) are allowed to communicate with the outside world and with database servers on the inside segment, but are not allowed to communicate with each other

For private VLANs to be applied, the switch must first be configured for standard VLAN operation. For more information, see Section 4.3.2, Standard VLAN Operation. When this is in place, one or more of the configured VLANs can be configured as private VLANs. Ports in a private VLAN fall into one of three groups:

- Promiscuous ports
- Ports from which traffic can be forwarded to all ports in the private VLAN
- Ports that can receive traffic from all ports in the private VLAN
- Community Ports
- Ports from which traffic can only be forwarded to community and promiscuous ports in the private VLAN
- Ports that can receive traffic from only community and promiscuous ports in the private VLAN
- Isolated ports
- Ports from which traffic can only be forwarded to promiscuous ports in the private VLAN
- Ports that can receive traffic from only promiscuous ports in the private VLAN

The configuration of promiscuous, community, and isolated ports applies to all private VLANs.

The forwarding of frames classified to a private VLAN happens:

- When traffic comes in on a promiscuous port in a private VLAN, the VLAN mask from the VLAN table is applied.
- When traffic comes in on a community port, the ISOLATED_PORT mask is applied in addition to the VLAN mask from the VLAN table.
- When traffic comes in on an isolated port, the ISOLATED_PORT mask and the COMMUNITY_PORT mask are applied in addition to the VLAN mask from the VLAN table.

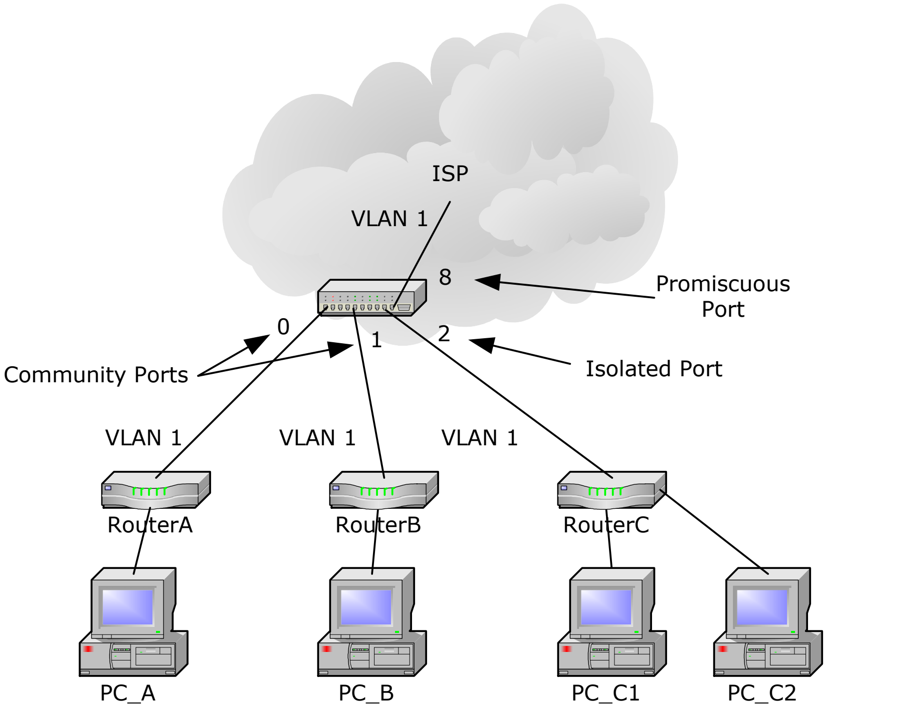
4.3.4.1 ISP Example

Situation:

Customers A, B, and C are connected to the same switch at the ISP. Customers A and B are allowed to communicate with each other, as well as the ISP. Customer C can only communicate with the ISP. VLAN 1 is the private VLAN that isolates Customers A, B from C. Traffic on VLAN 1 coming in from the ISP (port 8) uses the VLAN mask in the VLAN table. Traffic on VLAN 1 from customer A or B has the ISOLATED_PORTS mask applied in addition to the mask from the VLAN table, with the result that traffic from customer A and B is not forwarded to customers C. Traffic on VLAN 1 from customer C has the ISOLATED_PORTS mask and the COMMUNITY_PORTS mask applied in addition to the mask from the VLAN table, with the result that traffic from customer C is not forwarded to customers A and B.

The following illustration shows the desired setup.

FIGURE 4-2: ISP EXAMPLE FOR PRIVATE VLAN



VLAN Table

VID	Ports in Mask	Private
⋮	⋮	⋮
1	0, 1, 2, 8	1
⋮	⋮	⋮

ISOLATED_PORTS
(Isolated ports set to 0)

Bit	Value
0	1
1	1
2	0
⋮	⋮
8	1
⋮	⋮

COMMUNITY_PORTS
(Community ports set to 0)

Bit	Value
0	0
1	0
2	1
⋮	⋮
8	1
⋮	⋮

Resolution:

```
# It is assumed that Port VID and tag handling for VLAN 1 is already
# configured according to the description in Standard VLAN Operation.
# Configure VLAN 1 as a private VLAN in the VLAN table by performing these steps:
# - Point to VLAN 1.
# - Set it as private.
# - Disable mirroring of the VLAN (not important for the example).
# - Enable learning within the VLAN (not important for the example).
# - Disable source check within the VLAN (not important for the example).
# - Include ports 0, 1, 2, and 8 in the VLAN mask.
# Insert the entry into the VLAN table.
VLANTIDX.INDEX = 1
VLANTIDX.VLAN_PRIV_VLAN = 1
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0 (don’t care, for this example)
VLANTIDX.VLAN_SRC_CHK = 0 (don't care, for this example)
VLANACCESS.VLAN_PORT_MASK = 0x00000107
VLANACCESS.VLAN_TBL_CMD = 2
# Configure the private VLAN mask so that port 8 is a promiscuous
# port, ports 0 and 1 are community ports, and port 2 is an isolated port.
ISOLATED_PORTS.ISOL_PORTS = 0x00000103
COMMUNITY_PORTS.COMM_PORTS = 0x00000104
```

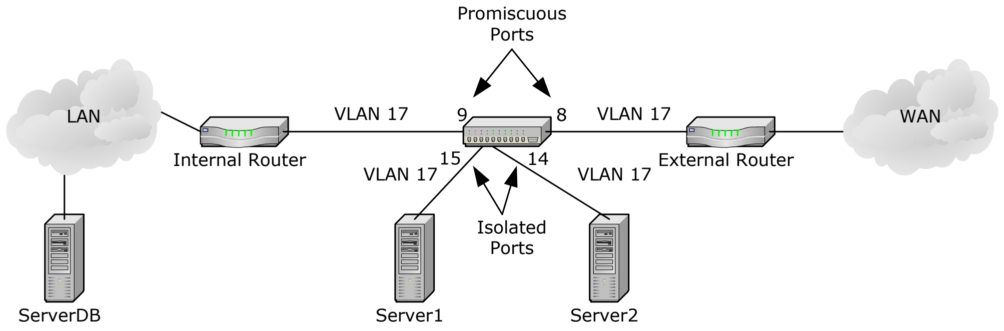
4.3.4.2 DMZ Example

Situation:

VLAN 17 is a private VLAN that isolates Server1 and Server2. Traffic on VLAN 17 coming from the internal or the external router (ports 8 and 9) uses the VLAN mask in the VLAN table. Traffic on VLAN 17 from Server1 and Server2 (ports 14 and 15) has the ISOLATED_PORTS applied in addition to the mask from the VLAN table, with the result that traffic from Server1 is not forwarded to Server2 and visa versa.

The following illustration shows the desired setup.

FIGURE 4-3: DMZ EXAMPLE FOR PRIVATE VLAN



VLAN Table

VID	Ports in Mask	Private
⋮	⋮	⋮
17	8, 9, 14, 15	1
⋮	⋮	⋮

ISOLATED_PORTS
(Promiscuous Ports Set to 1)

Bit	Value
⋮	⋮
8	1
9	1
⋮	⋮
14	0
15	0
⋮	⋮

Resolution:

```
# It is assumed that Port VID and tag handling for VLAN 17 is already
# configured according to the description in Standard VLAN Operation.
# Configure VLAN 17 as a private VLAN in the VLAN table by performing these steps:
# - Point to VLAN 17.
# - Set it as private.
# - Disable mirroring of the VLAN (not important for the example).
# - Enable learning within the VLAN (not important for the example).
# - Disable source check within the VLAN (not important for the example).
# - Include ports 8, 9, 14, and 15 in the VLAN mask.
# - Insert the entry into the VLAN table.
VLANTIDX.INDEX = 17
VLANTIDX.VLAN_PRIV_VLAN = 1
VLANTIDX.VLAN_MIRROR = 0 (don't care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0 (don't care, for this example)
VLANTIDX.VLAN_SRC_CHK = 0 (don't care, for this example)
VLANACCESS.VLAN_PORT_MASK = 0x0000C300
VLANACCESS.VLAN_TBL_CMD = 2
# Configure the private VLAN mask so that ports 8 and 9 are promiscuous
# ports.
ISOLATED_PORTS.ISOL_PORTS = 0x00000300
```

4.3.5 ASYMMETRIC VLANS

Asymmetric VLANs use the same configuration registers as for standard VLAN operation. For more information about standard VLAN operation, see Section 4.3.2, Standard VLAN Operation.

Asymmetric VLANs can be used to prevent communication between hosts in a network. This behavior is similar to what can be obtained by using private VLANs. For more information, seeSection 4.3.4, Private VLANs.

Situation:

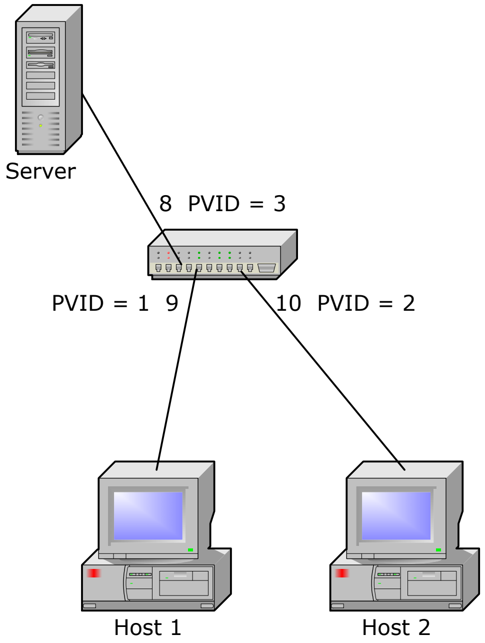
A server and two hosts are connected to a switch. Communication between the hosts and the server should be allowed, but the hosts are not allowed to communicate directly. All traffic between the server and the hosts is untagged. Host 1 is connected to port 9, host 2 to port 10, and the server to port 8.

The host-1 port gets port VID 1 and the host-2 port gets port VID 2. The server port is a member of both VLANs 1 and 2. The server port gets port VID 3, and the two host ports are members of VLAN 3, as shown in the following illustration.

FIGURE 4-4: ASYMMETRIC VLANS

VLAN Table

VID	Ports in Mask
⋮	⋮
1	8
2	8
3	9, 10
⋮	⋮



Resolution:

```
# Analyzer configurations common for ports 8, 9, and 10.
# Allow only untagged frames.
DROP_CFG.DROP_UNTAGGED_ENA = 0
DROP_CFG.DROP_C_TAGGED_ENA = 1
DROP_CFG.DROP_PRIO_C_TAGGED_ENA = 1
# As tagged frames are dropped all frames are classified to the port VID.
VLAN_CFG.VLAN_AWARE_ENA = 0 (don’t care, for this example)
# Configure the egress side of the port to not insert tags.
TAG_CFG.TAG_CFG = 0
# Analyzer configuration specific for port 8. Set the port VID to 3.
VLAN_CFG.VLAN_VID = 3
VLAN_CFG.VLAN_DEI = 0 (don’t care, for this example)
# Analyzer configuration specific for port 9. Set the port VID to 1.
VLAN_CFG.VLAN_VID = 1
VLAN_CFG.VLAN_DEI = 0 (don’t care, for this example)


# Analyzer configuration specific for port 10. Set the port VID to 2.
VLAN_CFG.VLAN_VID = 2
VLAN_CFG.VLAN_DEI = 0 (don’t care, for this example)
# Analyzer configuration common to all ports.
# Configure VLAN 1 to contain port 8.
VLANTIDX.INDEX = 1
VLANTIDX.VLAN_PRIV_VLAN = 0
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0
VLANTIDX.VLAN_SRC_CHK = 0
VLANACCESS.VLAN_PORT_MASK = 0x00000100
VLANACCESS.VLAN_TBL_CMD = 2
```

```
# Configure VLAN 2 to contain port 8.
VLANTIDX.INDEX = 2
VLANTIDX.VLAN_PRIV_VLAN = 0
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0
VLANTIDX.VLAN_SRC_CHK = 0
VLANACCESS.VLAN_PORT_MASK = 0x00000100
VLANACCESS.VLAN_TBL_CMD = 2
# Configure VLAN 3 to contain ports 9 and 10.
VLANTIDX.INDEX = 3
VLANTIDX.VLAN_PRIV_VLAN = 0
VLANTIDX.VLAN_MIRROR = 0 (don’t care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0
VLANTIDX.VLAN_SRC_CHK = 0
VLANACCESS.VLAN_PORT_MASK = 0x00000600
VLANACCESS.VLAN_TBL_CMD = 2
```

4.3.6 SPANNING TREE PROTOCOLS

This section provides information about Rapid Spanning Tree Protocol (RSTP) support and Multiple Spanning Tree Protocol (MSTP) support. The device also supports legacy Spanning Tree Protocol (STP). STP was obsoleted by RSTP in IEEE 802.1D and is not described in this document.

It is assumed that only LAN ports connected to the switch core participate in the spanning tree protocol. This implies that BPDUs are terminated by the switch core.

4.3.6.1 Rapid Spanning Tree Protocol

The following table lists the analyzer configuration registers for Rapid Spanning Tree Protocol (RSTP) operation.

TABLE 4-16: ANALYZER CONFIGURATIONS FOR RSTP SUPPORT

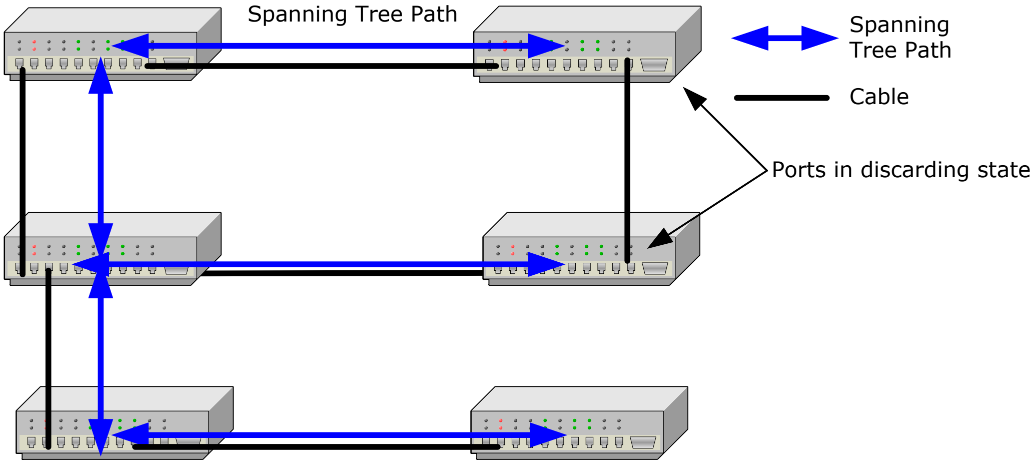
Register/Register Field	Description	Replication
PGID[80-91]	Source masks used for ingress filtering.	Per port
PGID[64-79]	Aggregation masks that can be used for egress filtering for RSTP.	16
PORT_CFG.LEARN_ENA	Enable learning per port.	Per port
CPU_FWD_BPDU_CFG	Enable redirection of frames with reserved BPDU DMAC addresses.	Per port per address
CPUQ_8021_CFG.CPUQ_BPDU_VAL	CPU extraction queue for redirected BPDU frames.	Per address

To eliminate potential loops in a network, the Rapid Spanning Tree Protocol in IEEE 802.1D creates a single path between any two bridges in a network, adding stability and predictability to the network. The protocol is implemented by assigning states to all ports. Each state controls a port’s functionality, limiting its ability to receive and transmit frames and learn addresses.

Establishing a spanning tree is done through the exchange of BPDUs between bridge entities. BPDUs are frequently exchanged between neighboring bridges. These frames are identified by the Bridge protocol address range (DMAC = 01-80-C2-00-00-0x).

When there is a change in the network topology, the protocol reconfigures the port states.

FIGURE 4-5: SPANNING TREE EXAMPLE



The following table lists the Rapid Spanning Tree port state properties.

TABLE 4-17: RSTP PORT STATE PROPERTIES

State	BPDU Reception	BPDU Generation	Frame Forwarding	SMAC Learning
Discarding	Yes	Yes	No	No
Learning	Yes	Yes	No	Yes
Forwarding	Yes	Yes	Yes	Yes

The legacy STP states disabled, blocking, and listening correspond to the discarding state of RSTP.

All frames with a Bridge protocol address must be redirected to the CPU. This is configured in CPU_FWD_BPDU_CFG. BPDUs are forwarded to the CPU irrespective of the port's RSTP state. CPUQ_8021_CFG.CPUQ_BPDU_VAL can be used to configure in which CPU extraction queue the BPDUs are placed. BPDU generation is done through frame injection from the CPU.

Frame forwarding is controlled through ingress filtering and egress filtering. Ingress filtering can be done by using the source masks (PGID[80-91]), and egress filtering can be done by using the aggregation masks (PGID[64-79]). Forwarding can be disabled for ports not in the Forwarding state by clearing their source masks and excluding them from all aggregation masks. The use of the aggregation masks for egress filtering does not preclude the combination of link aggregation and RSTP support. All ports in a link aggregation group that are not in the Forwarding state must be disabled in all aggregation masks. For link aggregated ports in the Forwarding state, the aggregation masks must be configured for link aggregation (such as when RSTP is not supported.)

Learning can be enabled per port with the PORT_CFG.LEARN_ENA.

The following table provides an overview of the port state configurations for port p.

TABLE 4-18: RSTP PORT STATE CONFIGURATION FOR PORT P

State	CPU_FWD_BPDU_CFG[p].BPDU_REDIR_ENA[0]	PGID[80+p]	PGID[64-79], All 16 Masks, Bit p	PORT_CFG[p].LEARN_ENA
Discarding	1	0	0	0
Learning	1	0	0	1
Forwarding	1	1 except for bit p	1	1

4.3.6.1.1 RSTP Example

Situation:

Port 0 is in the RSTP Discarding state. Port 2 is in the RSTP Learning state. Port 3 is in the RSTP Forwarding state. All other ports on the switch are unused.

Resolution:

```
# Get Spanning Tree Protocol BDPUs to CPU extraction queue 0 for port 0, 2, and 3.
CPU_FWD_BPDU_CFG[0].BPDU_REDIR_ENA[0] = 1
CPU_FWD_BPDU_CFG[2].BPDU_REDIR_ENA[0] = 1
CPU_FWD_BPDU_CFG[3].BPDU_REDIR_ENA[0] = 1
CPUQ_8021_CFG.CPUQ_BPDU_VAL[0] = 0
# Configure the source mask for port 0 (Discarding state).
PGID[80] = 0x00
# Configure the source mask for port 2 (Learning state).
PGID[82] = 0x00
# Configure the source mask for port 3 (Forwarding state).
PGID[83] = 0x77
# Configure the aggregation masks to only allow forwarding to port 3
# (Forwarding state).
PGID[64-79] = 0x08
# Configure the learn mask to only allow learning on ports
# 2 (Learning state) and 3 (Forwarding state).
PORT_CFG[0].LEARN_ENA = 0
PORT_CFG[2].LEARN_ENA = 1
PORT_CFG[3].LEARN_ENA = 1
```

4.3.6.2 Multiple Spanning Tree Protocol

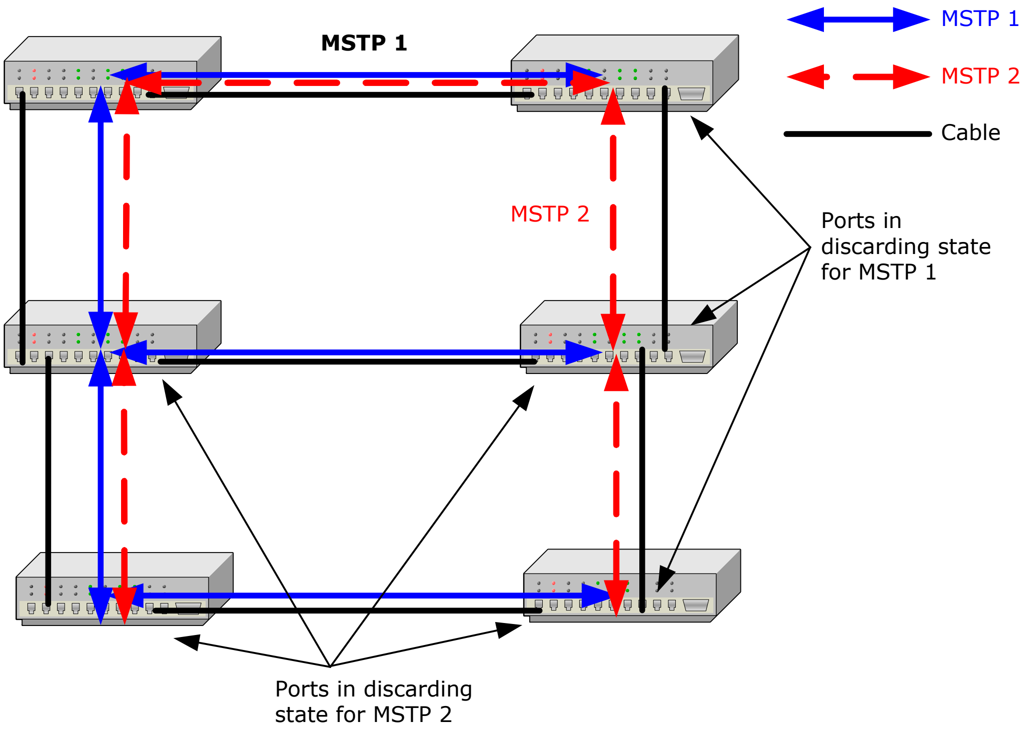
The following table lists the analyzer configuration registers for Multiple Spanning Tree Protocol (MSTP) operation.

TABLE 4-19: ANALYZER CONFIGURATIONS FOR MSTP SUPPORT

Register/Register Field	Description	Replication
VLANACCESS.VLAN_SRC_CHK	Per-VLAN ingress filtering enable. Part of VLAN table command for indirect access to configuration of the 4095 VLANs.	None
VLANMASK	Per-port VLAN ingress filtering enable.	None
ADVLEARN.VLAN_CHK	Disable learning for frames discarded due to VLAN membership source port filtering.	None
PORT_CFG.LEARN_ENA	Enable learning per port.	Per port
CPU_FWD_BPDU_CFG	Enable redirection of frames with reserved BPDU DMAC addresses.	Per port per address
CPUQ_8021_CFG.CPUQ_BPDU_VAL	CPU extraction queue for redirected BPDU frames.	Per address

The Multiple Spanning Tree Protocol (MSTP) in IEEE 802.1Q increases network use, relative to RSTP, by creating multiple spanning trees that VLANs can map to independently, rather than having only one path between bridges common for all VLANs. The multiple spanning trees are created by assigning different bridge identifiers for each spanning tree. Mapping the VLANs to spanning trees is done arbitrarily.

FIGURE 4-6: MULTIPLE SPANNING TREE EXAMPLE


MSTP 1
MSTP 1
MSTP 2
Cable
MSTP 2
Ports in discarding state for MSTP 1
Ports in discarding state for MSTP 2

The Learning state is not supported for MSTP. However, this has limited impact, because when the port is taken to the Forwarding state, learning is done at wire-speed, and, as a result, the SMAC learn delay is less important. MSTP is supported for all VLANs.

The following table lists the multiple spanning tree port state properties.

TABLE 4-20: MSTP PORT STATE PROPERTIES

State per VLAN	BPDU Reception	BPDU Generation	Frame Forwarding	SMAC Learning
Discarding	Yes	Yes	No	No
Learning (not supported)	Yes	Yes	No	Yes
Forwarding	Yes	Yes	Yes	Yes

To enable the MSTP port states:

- Ensure that the switch is VLAN-aware. For more information, see Section 4.3.2, Standard VLAN Operation.
- Set the ADVLEARN.VLAN_CHK bit to prevent learning of frames discarded due to VLAN ingress filtering.
- Configure all ports as defined for the forwarding state of the RSTP port. For more information, see Table 4-18.

Port states per VLAN are hereafter solely configured through the VLAN masks as listed in the following table for port p and VLAN v.

TABLE 4-21: MSTP PORT STATE CONFIGURATION FOR PORT P AND VLAN V

State	VLAN_ACCESS. VLAN_SRC_CHKVLAN v	VLAN_ACCESS. VLAN_PORT_MASK Bit p, VLAN v
Discarding	1	0
Learning	Not supported	Not supported
Forwarding	1	1

As an alternative to setting the VLANACCESS.`VLAN_SRC_CHK` bit in all VLAN entries in the VLAN table, VLAN ingress filtering can be enabled globally for all VLANs on a per port basis through VLANMASK.

For all multiple spanning tree instances, BPDUs are forwarded to the CPU irrespective of the port states.

4.3.6.2.1 MSTP Example

Situation:

Ports 10 and 11 are both members of VLANs 20 and 21. Two spanning trees are used:

- Spanning tree for VLAN 20, where both ports 10 and 11 are in the Forwarding state
- Spanning tree for VLAN 21, where port 10 is in the Discarding state and port 11 is in the Forwarding state

All other ports on the switch are unused.

Resolution:

```
# Get all BDPUs to CPU queue 0.
CPU_FWD_BPDU_CFG[*].BPDU_REDIR_ENA[0] = 1
CPUQ_8021_CFG.CPUQ_BPDU_VAL[0] = 0
# Enable learning on all ports. The VLAN table controls forwarding and learning.
DEV::PORT_CFG.LEARN_ENA = 1
# Disable learning of VLAN membership source port filtered frames.
ADVLEARN.VLAN_CHK = 1
# Configure VLAN 20 for ports 10 and 11 in Forwarding state.
VLANTIDX.INDEX = 20
VLANTIDX.VLAN_PRIV_VLAN = 0
VLANTIDX.VLAN_MIRROR = 0 (don't care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0
VLANTIDX.VLAN_SRC_CHK = 1
VLANACCESS.VLAN_PORT_MASK = 0x00000C00
VLANACCESS.VLAN_TBL_CMD = 2
# Configure VLAN 21 for port 10 in Discarding state and port 11 in Forwarding state.
VLANTIDX.INDEX = 21
VLANTIDX.VLAN_PRIV_VLAN = 0
VLANTIDX.VLAN_MIRROR = 0 (don't care, for this example)
VLANTIDX.VLAN_LEARN_DISABLE = 0
VLANTIDX.VLAN_SRC_CHK = 1
VLANACCESS.VLAN_PORT_MASK = 0x00000800
VLANACCESS.VLAN_TBL_CMD = 2
```

4.3.7 IEEE 802.1X: NETWORK ACCESS CONTROL

IEEE 802.1X, Port-Based Network Access Control, provides a standard for authenticating and authorizing devices attached to a LAN port.

Generally, IEEE 802.1X is port-based; however, the device also supports MAC-based network access control.

This section provides information about the configuration settings for port-based and MAC-based network access control.

4.3.7.1 Port-Based Network Access Control

The following table lists the configuration settings required for port-based network access control.

TABLE 4-22: CONFIGURATIONS FOR PORT-BASED NETWORK ACCESS CONTROL

Register/Register Field	Description / Value	Replication
ANA::CPU_FWD_BPDU_ CFG.BPDU_REDIR_ENA[3]	Must be set to 1 to redirect frames with destination MAC addresses 01-80-C2-00-00-03 to the CPU Port Module. IEEE 802.1X uses MAC address 01-80-C2-00-00-03.	Per port
ANA::CPUQ_8021_CFG. CPUQ_BPDU_VAL[3]	Queue to which authentication BPDUs are redirected.	None
ANA::PGID[64-79]	When a port is not yet authenticated, any forwarding of frames to the port can be disabled by clearing the port's bit in all 16 aggregation masks. After authenticated, these bits must be set.	16

TABLE 4-22: CONFIGURATIONS FOR PORT-BASED NETWORK ACCESS CONTROL

Register/Register Field	Description / Value	Replication
ANA::PGID[80-91]	Source masks. When a port is not yet authenticated, any forwarding of frames received on the port must be disabled. This can be done by setting the ANA::PGID[80+port] to all-zeros. After authentication, the port's source mask must be set back to its normal value.	Per port

The configuration settings required for port-based network access control enable the following functionality:

- Redirects frames with DMAC 01-80-C2-00-00-03 to CPU, even if the port is not yet authenticated.
- Stops forwarding of frames to ports that are not yet authenticated. This is configured in ANA::PGID[64-79].
- Stops forwarding of frames received on ports that are not yet authenticated. This is configured in ANA::PGID[80-91].

4.3.7.2 MAC-Based Authentication with Secure CPU-Based Learning

The following table lists the configuration settings required for MAC-based network access control with secure CPU-based learning.

TABLE 4-23: CONFIGURATIONS FOR MAC-BASED NETWORK ACCESS CONTROL WITH SECURE CPU-BASED LEARNING

Register/Register Field	Description / Value	Replication
ANA:PORT:CPU_FWD_BPDU_CFG.BPDU_REDIR_ENA[3]	Must be set to 1 to redirect frames with destination MAC addresses 01-80-C2-00-00-03 to the CPU Port Module. IEEE 802.1X uses MAC address 01-80-C2-00-00-03.	Per port
ANA::CPUQ_8021_CFG.CPUQ_BPDU_VAL[3]	Queue to which authentication BPDUs are redirected.	None

ANA:PORT:PORT_CFG.LEARN_ENA
ANA:PORT:PORT_CFG.LEARNCPU
ANA:PORT:PORT_CFG.LEARNDROP
ANA:PORT:PORT_CFG.LEARNAUTO
	Must be set to support secure CPU-based learning. See Section 4.3.1.2, Address Learning.
	PORT_CFG.LEARN_ENA = 1
	PORT_CFG.LEARNCPU = 1
	PORT_CFG.LEARNDROP = 1
	PORT_CFG.LEARNAUTO = 0
		Per port

The MAC-based network access control with secure CPU-based learning enables the following functionality:

- Redirects frames with DMAC 01-80-C2-00-00-03 to CPU.
- Only frames from known, authenticated MAC addresses are forwarded to other ports.
- Frames from unknown MAC addresses are redirected to CPU for authentication. After the address is authenticated, the CPU must insert an entry in the MAC table. The authentication process may be initiated from the CPU when receiving learn frames.

4.3.7.3 MAC-Based Authentication with No Learning

The following table lists the configuration settings required for MAC-based network access control with no learning.

TABLE 4-24: CONFIGURATIONS FOR MAC-BASED NETWORK ACCESS CONTROL WITH NO LEARNING

Register/Register Field	Description / Value	Replication
ANA:PORT:CPU_FWD_BPDU_CFG.BPDU_REDIR_ENA[3]	Must be set to 1 to redirect frames with destination MAC addresses 01-80-C2-00-00-03 to the CPU Port Module. IEEE 802.1X uses MAC address 01-80-C2-00-00-03.	Per port
ANA::CPUQ_8021_CFG.CPUQ_BPDU_VAL[3]	Queue to which authentication BPDUs are redirected.	None

ANA:PORT:PORT_CFG.LEARN_ENA
ANA:PORT:PORT_CFG.LEARNCPU
ANA:PORT:PORT_CFG.LEARNDROP
ANA:PORT:PORT_CFG.LEARNAUTO
	Must be set to support no learning. See Section 4.3.1.2, Address Learning.
	PORT_CFG.LEARN_ENA = 1
	PORT_CFG.LEARNCPU = 0
	PORT_CFG.LEARNDROP = 1
	PORT_CFG.LEARNAUTO = 0
		None

The MAC-based network access control with no learning enables the following functionality:

- Frames with `DMAC 01-80-C2-00-00-03` are redirected to CPU. Unauthenticated and unauthorized devices must initiate an 802.1X session by sending 802.1X BPDUs (MAC address: `01-80-C2-00-00-03`). After the address is authenticated, the CPU must insert an entry in the MAC table.
- Only frames from known, authenticated MAC addresses are forwarded to other ports.
- Frames from unknown MAC addresses are discarded and the CPU can therefore not initiate the authentication process.

4.3.8 LINK AGGREGATION

Link aggregation bundles multiple ports (member ports) together into a single logical link. It is primarily used to increase available bandwidth without introducing loops in the network and to improve resilience against faults. A link aggregation group (LAG) can be established with individual links being dynamically added or removed. This enables bandwidth to be incrementally scaled based on changing requirements. A link aggregation group can be quickly reconfigured if faults are identified.

Frames destined for a LAG are sent on only one of the LAG's member ports. The member port on which a frame is forwarded is determined by a 4-bit aggregation code (AC) that is calculated for the frame.

The aggregation code ensures that frames belonging to the same frame flow (for example, a TCP connection) are always forwarded on the same LAG member port. For that reason, reordering of frames within a flow is not possible. The aggregation code is based on the following information:

- SMAC
- DMAC
- Source and destination IPv4 address.
- Source and destination TCP/UDP ports for IPv4 packets
- Source and destination TCP/UDP ports for IPv6 packets
- IPv6 Flow Label

For best traffic distribution among the LAG member ports, enable all contributions to the aggregation code.

Each LAG can consist of up to 16 member ports. Any quantity of LAGs may be configured for the device (only limited by the quantity of ports on the device.) To configure a proper traffic distribution, the ports within a LAG must use the same link speed.

A port cannot be a member of multiple LAGs.

4.3.8.1 Link Aggregation Configuration

The following table lists the registers associated with link aggregation groups.

TABLE 4-25: LINK AGGREGATION GROUP CONFIGURATION REGISTERS

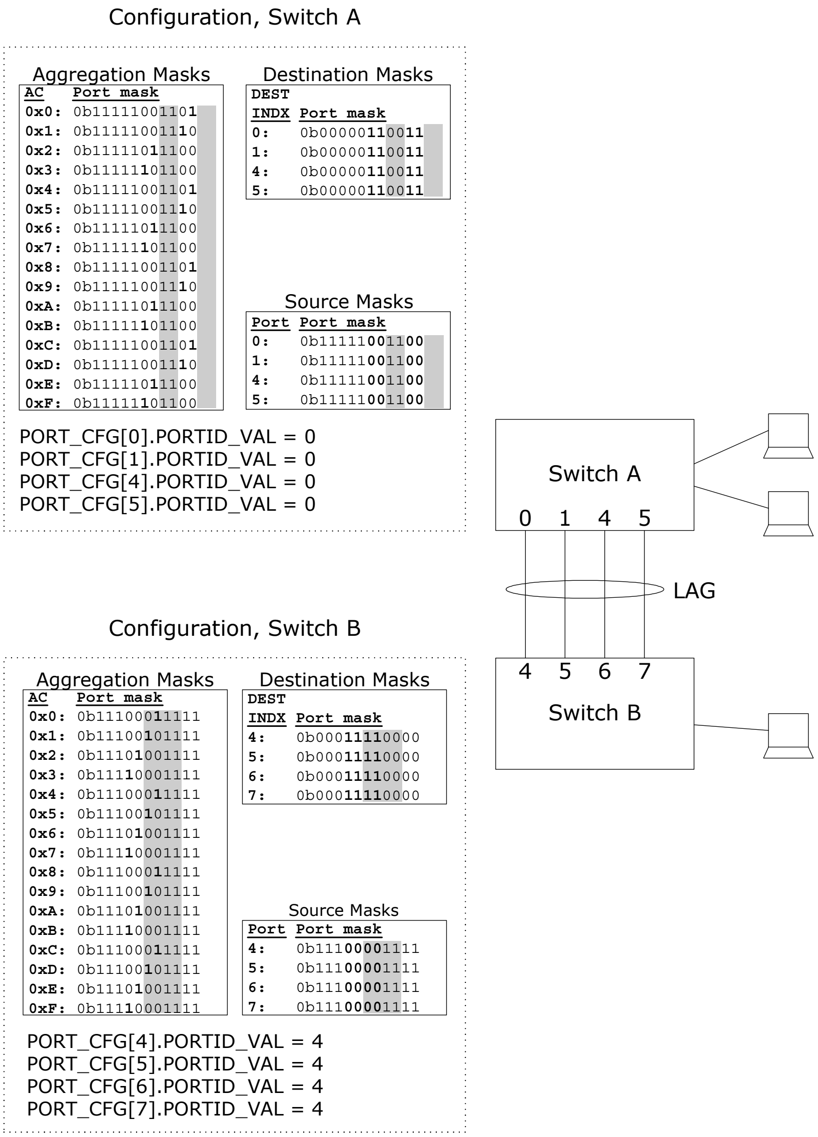
Register/Register Field	Description / Value	Replication
ANA::PGID[0 – 63]	Destination mask	64
ANA::PGID[80 – 91]	Source mask.	Per port
ANA::PGID[64 – 79]	Aggregation mask.	16
ANA::PORT_CFG.PORTID_VAL	Logical port number. Must be set to the same value for all ports that are part of a given LAG; for example, the lowest port number that is a member of the LAG.	Per port
ANA::AGGR_CFG. AC_IP6_FLOW_LBL_ENA	Use IPv6 flow label when calculating AC. Configure identically for all ports. Recommended value is 1.	None
ANA::AGGR_CFG. AC_IP4_SIPDIP_ENA	Use IPv4 source and destination IP address when calculating aggregation code. Configure identically for all ports. Recommended value is 1.	None
ANA::AGGR_CFG.AC_IP4_TCPUDP_PORT_ENA	Use IPv4 TCP/UDP port when calculating aggregation code. Configure identically for all ports. Recommended value is 1.	None
ANA::AGGR_CFG.AC_IP6_TCPUDP_PORT_ENA	Use IPv6 TCP/UDP port when calculating aggregation code. Configure identically for all ports. Recommended value is 1.	None
ANA:: AGGR_CFG. AC_DMAC_ENA	Use destination MAC address when calculating aggregation code. Configure identically for all ports. Recommended value is 1.	None
ANA:: AGGR_CFG. AC_SMAC_ENA	Use source MAC address when calculating aggregation code. Configure identically for all ports. Recommended value is 1.	None
ANA:: AGGR_CFG. AC_RND_ENA	Use random aggregation code. Recommended value is 0.	None

To set up a link aggregation group, the following destination masks, source masks, and aggregation masks must be configured:

- **Destination Masks: ANA::PGID[0-63]**. For each of the member ports, the corresponding destination mask must be configured to include all member ports of the LAG.
- **Source Masks: ANA::PGID[80-91]**. The source masks must be configured to avoid flooding frames that are received at one member port back to another member port of the LAG. As a result, the source masks for each of the member ports must be configured to exclude all of the LAG’s member ports.
- **Aggregation Masks: ANA::PGID[64-79]**. The aggregation masks must be configured to ensure that when a frame is destined for the LAG, it gets forwarded to exactly one of the LAG’s member ports. Also, the distribution of traffic between member ports is determined by this configuration.

The following illustration shows an example of a LAG configuration.

FIGURE 4-7: LINK AGGREGATION EXAMPLE



Configuration, Switch A

Aggregation Masks

AC	Port mask
0x0:	0b1111100110**1**
0x1:	0b111110011**1**0
0x2:	0b111110**1**1100
0x3:	0b11111**1**01100
0x4:	0b1111100110**1**
0x5:	0b111110011**1**0
0x6:	0b111110**1**1100
0x7:	0b11111**1**01100
0x8:	0b1111100110**1**
0x9:	0b111110011**1**0
0xA:	0b111110**1**1100
0xB:	0b11111**1**01100
0xC:	0b1111100110**1**
0xD:	0b111110011**1**0
0xE:	0b111110**1**1100
0xF:	0b11111**1**01100

Destination Masks

DEST INDX	Port mask
0:	0b00000**11**00**11**
1:	0b00000**11**00**11**
4:	0b00000**11**00**11**
5:	0b00000**11**00**11**

Source Masks

Port	Port mask
0:	0b11111**00**11**00**
1:	0b11111**00**11**00**
4:	0b11111**00**11**00**
5:	0b11111**00**11**00**

PORT_CFG[0].PORTID_VAL = 0
PORT_CFG[1].PORTID_VAL = 0
PORT_CFG[4].PORTID_VAL = 0
PORT_CFG[5].PORTID_VAL = 0

Configuration, Switch B

Aggregation Masks

AC	Port mask
0x0:	0b111000**1**1111
0x1:	0b11100**1**01111
0x2:	0b1110**1**001111
0x3:	0b111**1**0001111
0x4:	0b111000**1**1111
0x5:	0b11100**1**01111
0x6:	0b1110**1**001111
0x7:	0b111**1**0001111
0x8:	0b111000**1**1111
0x9:	0b11100**1**01111
0xA:	0b1110**1**001111
0xB:	0b111**1**0001111
0xC:	0b111000**1**1111
0xD:	0b11100**1**01111
0xE:	0b1110**1**001111
0xF:	0b111**1**0001111

Destination Masks

DEST INDX	Port mask
4:	0b000**1111**0000
5:	0b000**1111**0000
6:	0b000**1111**0000
7:	0b000**1111**0000

Source Masks

Port	Port mask
4:	0b111**0000**1111
5:	0b111**0000**1111
6:	0b111**0000**1111
7:	0b111**0000**1111

PORT_CFG[4].PORTID_VAL = 4
PORT_CFG[5].PORTID_VAL = 4
PORT_CFG[6].PORTID_VAL = 4
PORT_CFG[7].PORTID_VAL = 4

In this example, ports 0, 1, 4, and 5 of switch A are configured as a LAG. These ports are connected to 4 ports (4, 5, 6, 7) of switch B, providing an aggregated bandwidth of 4 Gbps between the two switches.

The aggregation masks for switch A are configured such that frames (destined for the LAG) are distributed on the member ports as follows:

- Port 0 if frame's aggregation code (AC) is 0x0, 0x4, 0x8, 0xC
- Port 1 if frame's aggregation code (AC) is 0x1, 0x5, 0x9, 0xD
- Port 4 if frame's aggregation code (AC) is 0x2, 0x6, 0xA, 0xE
- Port 5 if frame's aggregation code (AC) is 0x3, 0x7, 0xB, 0xF

4.3.8.2 Link Aggregation Control Protocol (LACP)

LACP allows switches connected to each other to automatically discover if any ports are member of the same LAG.

To implement LACP, any LACP frames must be redirected to the CPU. Such frames are identified by the DMAC being equal to 01-80-C2-00-00-02 (Slow Protocols Multicast address).

The following table lists the registers associated with configuring the redirection of LACP frames to the CPU.

TABLE 4-26: CONFIGURATION REGISTERS FOR LACP FRAME REDIRECTION TO THE CPU

Register/Register Field	Description / Value	Replication
ANA::CPU_FWD_BPDU_CFG.BPDU_REDIR_ENA[2]	Must be set to 1.	Per port

4.3.9 SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)

This section provides information about the port module registers and the analyzer registers for SNMP operation.

The following table lists the system registers for SNMP operation.

TABLE 4-27: SYSTEM REGISTERS FOR SNMP SUPPORT

Register	Description	Replication
CNT	The value of the counter. For more information about how to read counters, see Section 2.6, Statistics.	None

The following table lists the analyzer registers for SNMP support.

TABLE 4-28: ANALYZER REGISTERS FOR SNMP SUPPORT

Register	Description	Replication
MACACCESS	Command register for indirect MAC table access. Supports `GET_NEXT` command.	None
MACHDATA	High part of data word when accessing MAC table.	None
MACLDATA	Low part of data word when accessing MAC table.	None
MACTINDX	Index for direct-mode access to MAC table.	None

For SNMP support according to IETF RFC 1157, use the following features:

- RMON counters
- MAC table GET_NEXT function

For more information about the supported RMON counters, see Section 4.2.2, Port Counters.

For more information about the MAC table GET_NEXT function, see Table 2-74.

4.3.10 MIRRORING

To debug network problems, selected traffic can be copied, or mirrored, to a mirror port where a frame analyzer can be attached to analyze the frame flow.

The traffic to be copied to the mirror port can be selected as follows:

- All frames received on a given port (also known as ingress mirroring)
- All frames transmitted on a given port (also known as egress mirroring
- Frames selected through configured VCAP entries
- All frames classified to specific VIDs
- All frames sent to the CPU (may be useful for software debugging)
- Frames where the source MAC address is to be learned (also known as learn frame), which may be useful for software debugging

The mirror port may be any port on the device, including the CPU.

4.3.10.1 Mirroring Configuration

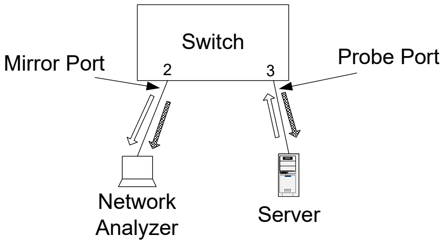
The following table lists configuration registers associated with mirroring.

TABLE 4-29: CONFIGURATION REGISTERS FOR MIRRORING

Register/Register Field	Description / Value	Replication
ANA::PORT_CFG.SRC_MIRROR_ENA	If set, all frames received on this port are mirrored to the port set configured in MIRRORPORTS, that is, ingress mirroring.	Per port
ANA:: EMIRRORMASK	Frames forwarded to ports in this mask are mirrored to the port set configured in MIRRORPORTS, that is, egress mirroring.	Per port
ANA::VLANTIDX.VLAN_MIRROR	If set, all frames classified to this VLAN are mirrored to the port set configured in MIRRORPORTS.	One per VID
ANA::AGENCTRL.MIRROR_CPU	Frames destined for the CPU extraction queues are also forwarded to the port set configured in MIRRORPORTS.	None
ANA::MIRRORPORTS	The mirror ports. Usually only one mirror port is configured, that is, only one bit is set in this mask.	None
ANA::CPUQ_CFG.CPUQ_MIRROR	CPU extraction queue used, if CPU is included in MIRRORPORTS.	None
ANA::ADVLEARN.LEARN_MIRROR	Learn frames are also forwarded to ports marked in MIRRORPORTS.	None
VCAP Registers	Configuration of VCAP entries, for example, to trigger copy to mirror port. For more information, see Section 2.8.1.2, VCAP IS2 Port Configuration.	Per VCAP entry

The following illustration shows a port mirroring example.

FIGURE 4-8: PORT MIRRORING EXAMPLE



All traffic to and from the server on port 3 (the probe port) is mirrored to port 2 (the mirror port). Note that the mirror port may become congested, because both the Rx frames and Tx frames on the probe port become Tx frames on the mirror port. The following mirror configuration is required:

```
ANA::PORT_CFG[3].SRC_MIRROR_ENA = 1
ANA::EMIRRORMASK[3] = 1
ANA::MIRRORPORTS = 0x0000004
```

In addition to the mirror configuration settings, the egress configuration of the mirror port (port 2) must be configured identically to the egress configuration of the probe port (port 3). This is to ensure that VLAN tagging and DSCP remarking at the mirror port is performed consistently with that of the probe port, such that the frame copies at the mirror port are identical to the original frames on the probe port.

Multiple mirror conditions, such as mirror multiple probe ports, VLANs, and so on, can be enabled concurrently to the same mirror port. However, in such configurations, it may not be possible to configure the egress part of the mirror port to perform tagging and DSCP remarking consistent with that of the original frame.

4.4 IGMP and MLD Snooping

This section provides information about the features and configurations related to Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) snooping.

By default, Layer-3 multicast data traffic is flooded in a Layer-2 network in the broadcast domain spanned by the VLAN. This causes unnecessary traffic in the network and extra processing of unsolicited frames in hosts not listening to the multicast traffic. IGMP and MLD snooping enables a Layer-2 switch to listen to IGMP and MLD conversations between host and routers. The switch can then prune multicast traffic from ports that do not have a multicast listener, and as a result, do not need a copy of the multicast frame. This is done by managing the multicast group addresses and the associated port masks.

IGMP is used to manage IPv4 multicast memberships, and MLD is used to manage IPv6 multicast memberships.

The device supports IGMPv2/v3 and MLDv1/v2. IGPMv2 and MLDv1 use any-source multicasting (ASM), where the multicast listener joins a group and can receive the multicast traffic from any source. IGMPv3 and MLDv2 introduce source-specific multicasting (SSM), where both source and group are specified by the multicast listener when joining a group.

The support in the device is two-fold:

- Control plane: IGMP and MLD frames are redirected to the CPU. This enables the CPU to listen to the queries and reports.
- Data plane: By monitoring the multicast group registrations and de-registrations signaled through the IGMP and MLD frames, the CPU can setup multicast group addresses and associated ports.

4.4.1 IGMP AND MLD SNOOPING CONFIGURATION

To implement IGMP and MLD snooping, any IGMP or MLD frames must be redirected to the CPU. For information about by the conditions by which such frames are identified, see Section 2.7.6, CPU Forwarding Determination. IGMP and MLD frames can be independently snooped and assigned individual CPU extraction queues.

The following table lists the registers associated with configuring the redirection of IGMP and MLD frames to the CPU.

TABLE 4-30: CONFIGURATION REGISTERS FOR IGMP AND MLD FRAME REDIRECTION TO CPU

Register/Register Field	Description / Value	Replication
ANA::CPU_FWD_CFG.IGMP_REDIR_ENA	Must be set to 1 to redirect IGMP frames to the CPU.	Per port
ANA::CPU_FWD_CFG.MLD_REDIR_ENA	Must be set to 1 to redirect MLD frames to the CPU.	Per port
ANA::CPUQ_CFG.CPUQ_IGMP	CPU extraction queue for IGMP frames.	None
ANA::CPUQ_CFG.CPUQ_MLD	CPU extraction queue for MLD frames.	None

4.4.2 IP MULTICAST FORWARDING CONFIGURATION

The following table lists the registers associated with configuring the multicast group addresses and the associated ports.

TABLE 4-31: IP MULTICAST CONFIGURATION REGISTERS

Register/Register Field	Description / Value	Replication
MACHDATA	MAC address and VID when accessing the MAC table.	None
MACLDATA	MAC address when accessing the MAC table.	None
MACTINDX	Direct address into the MAC table for direct read and write.	None
MACACCESS	Flags and command when accessing the MAC table.	None
MACTOPTIONS	Flags when accessing the MAC table	None
FLOODING_IPMC	Index into the PGID table used for flooding of IPv4/6 multicast control and data frames.	None

TABLE 4-31: IP MULTICAST CONFIGURATION REGISTERS (CONTINUED)

Register/Register Field	Description / Value	Replication
PGID[63:0]	Destination and flooding masks table.	64
IS1_ACTION.FID_SEL	Specifies the use of IS1_ACTION.FID_VAL for the DMAC lookup, the SMAC lookup, or for both lookups.	Per IS1 entry
IS1_ACTION.FID_VAL	FID value.	Per IS1 entry

IPv4 and IPv6 multicast group addresses are programmed in the MAC table as IPv4 and IPv6 multicast entries. For more information, see Section 2.9.1, MAC Table. The entry in the MAC table also holds the set of egress ports associated with the group address.

By default, programming an IPv4 or IPv6 multicast entry in the MAC table makes it an any-source multicast, because the actual source IP address is insignificant with respect to forwarding.

To create source-specific IPv4 or IPv6 multicast entries, the Filter Identifier (FID) action in VCAP IS1 can be used, which enables creation of specific FIDs per source IP address. Entries in IS1 can contain the full IPv4 or IPv6 source address. Multiple MAC table entries holding the same IPv4 or IPv6 multicast group address but different FIDs can then be created. This effectively enables source-specific multicasting.

The switch provides full control of flooding of unknown IP multicast frames. For more information, see Section 2.9.3.1, DMAC Analysis. Generally, an IGMP and MLD snooping switch disables flooding of unknown multicast frames, except to ports connecting to multicast routers. Note that unknown IPv4 multicast control frames should be flooded to all ports, because IPv4 is not as strict as IPv6 in terms of registration for IP multicast groups.

4.5 Quality of Service (QoS)

This section discusses features and configurations related to QoS.

The device includes a number of features related to providing low-latency guaranteed services to critical network traffic such as voice and video in contrast to best-effort traffic such as web traffic and file transfers.

All incoming frames are classified to a QoS class, which is used in the queue system when assigning resources, in the arbitration from ingress to egress queues and in the egress scheduler when selecting the next frame for transmission. The device provides two methods for classifying to a QoS class and for remarking priority information in the frame: Basic and Advanced classification.

Basic QoS classification enables predefined schemes for handling Priority Code Points (PCP), Drop Eligible Indicator (DEI), and Differentiated Service Code Points (DSCP):

- QoS classification based on PCP and DEI for tagged frames. The mapping table from PCP and DEI to QoS class is programmable per port.
- QoS classification based on DSCP values. Can optionally use only trusted DSCP values. The mapping table from DSCP value to QoS class is common between all ports.
- The device has the option to work as a DS boundary node connecting two DS domains together by translating incoming/outgoing DSCP values for selected ports.
- The DSCP values can optionally be remarked based on the frame’s classified QoS class.
- For untagged or non-IP frames, a default per-port QoS class is programmable.

Advanced QoS classification uses the VCAP IS1, which provides a flexible classification:

- A large range of higher layer protocol fields (Layer 2 through Layer 4) are available for rule matching.
- The IS1 action vector returns a QoS class, and translations of PCP, DEI, and DSCP values are also possible.
- Through programming of entries in IS1, QoS rules can be made as specific as needed.For example; per source MAC address, per TCP/UDP destination port number, or combination of both.

For more information about advanced QoS classification using the VCAP IS1, see Section 4.6.2, Ingress Control Lists.

4.5.1 BASIC QOS CONFIGURATION

The following table lists the registers associated with configuring basic QoS.

TABLE 4-32: BASIC QOS CONFIGURATION REGISTERS

Register	Description	Replication
ANA:PORT:QOS_CFG	QoS and DSCP configuration	Per port
ANA:PORT:QOS_PCP_DEI_MAP_CFG:	Mapping of DEI and PCP to QoS class and drop precedence level	Per port
ANA::DSCP_CFG	DSCP configuration	Per DSCP

Situation:

Assume a configuration with the following requirements:

- All frames with DSCP=7 must get QoS class 7.
- All frames with DSCP=8 must get QoS class 5.
- DSCP = 9 is untrusted and all frames with DSCP=9 should be treated as a non-IP frame.
- VLAN-tagged frames with PCP=7 must get QoS class 7
- All other IP frames must get QoS class 1.
- All other non-IP frames must get QoS class 0.

Solution:

```
# Program overall QoS configuration
QOS_CFG.QOS_DSCP_ENA = 1
QOS_CFG.QOS_PCP_ENA = 1


# Program DSCP trust configuration ("*" = 0 through 63)
DSCP_CFG[*].DSCP_TRUST_ENA = 1
DSCP_CFG[9].DSCP_TRUST_ENA = 0

# Program DSCP QoS configuration ("*" = 0 through 63)
DSCP_CFG[*].QOS_DSCP_VAL = 1
DSCP_CFG[7].QOS_DSCP_VAL = 7
DSCP_CFG[8].QOS_DSCP_VAL = 5

# Program PCP QoS configuration ("*" = 0 through 15)
# Note: both 7 and 15 are programmed in order to don't care DEI
QOS_PCP_DEI_MAP_CFG[*] = 0
QOS_PCP_DEI_MAP_CFG[7] = 7
QOS_PCP_DEI_MAP_CFG[15] = 7

# Program default QoS class for non-IP, non-tagged frames.
QOS_CFG.QOS_DEFAULT_VAL = 0
```

4.5.2 IPV4 AND IPV6 DSCP REMARKING

IPv4 and IPv6 packets include a 6-bit Differentiated Services Code Point (DSCP), which switches and routers can use to determine the QoS class of a frame. With a proper value in the DSCP field, packets can be prioritized consistently throughout the network. Compared to QoS classification based on user priority, classification based on DSCP provides two main advantages

- DSCP field is already present in all packets (assuming all traffic is IPv4/IPv6).
- DSCP value is preserved during routing and is therefore better suited for end-to-end QoS signaling.

Some hosts may be able to send packets with an appropriate value in the DSCP field, whereas other hosts may not provide an appropriate value in the DSCP field.

For packets without an appropriate value in the DSCP field, the device can be configured to write a new DSCP value into the frame, based on the QoS class of the frame. For example, the device may have determined the QoS class based on the VLAN tag priority information (PCP and DEI). After the packet is transmitted by the egress port, the DSCP field can be rewritten with a value based on the QoS class of the frame. Any subsequent routers or switches can then be easily prioritize the frame, based on the rewritten DSCP value.

The DSCP rewriting functionality available in the device provides flexible, per-ingress port and per-DSCP-value configuration of whether frames should be subject to DSCP rewrite. If it is determined at the ingress port that the DSCP value should be rewritten and to which value, this is then signaled to the egress ports, where the actual change of the DSCP field is done.

In addition, the IS1 can be programmed to return a DSCP value as part of the action vector. This value overrules the potential DSCP value coming out of the DSCP rewrite functionality described previously. A DSCP value from either the basic classification or the advanced IS1 classification obey the same egress rules for the actual DSCP remarking.

4.5.2.1 DSCP Remarking Configuration

The following table lists the configuration registers associated with DSCP remarking.

TABLE 4-33: CONFIGURATION REGISTERS FOR DSCP REMARKING

Register/Register Field	Description / Value	Replication
ANA:PORT:DSCP_REWR_CFG	Two-bit DSCP rewrite mode per ingress port. 0x0: No DSCP rewrite. 0x1: Rewrite only if the frame’s current DSCP value is zero. 0x2: Rewrite only if the frame’s current DSCP value is enabled for remarking in ANA::DSCP_CFG.DSCP_REWR_ENA. 0x3: Rewrite DSCP of all frames, regardless of current DSCP value.	Per ingress port
ANA::DSCP_CFG.DSCP_REWR_ENA	Enables specific DSCP values for rewrite for ports with DSCP rewrite mode set to 0x2.	Per DSCP
ANA::DSCP_REWR_CFG.DSCP_QOS_REWR_VAL	Maps the frame’s DP level and QoS class to a DSCP value.	Per DP level and per QoS class
REW::DSCP_CFG.DSCP_REWR_CFG	Enables DSCP rewrite for egress port.	Per egress port
REW::DSCP_REMAP_CFG	Remap table of DSCP values for drop precedence 0.	None
REW::DSCP_REMAP_DP_1_CFG	Remap table of DSCP values for drop precedence 1.	None

The configuration related to the ingress port controls whether a frame is to be remarked. For each ingress port, a DSCP rewrite mode is configured in ANA:PORT:DSCP_REWR_CFG. This register defines the four different modes as follows:

- 0x0: No DSCP rewrite, that is, never change the received DSCP value.
- 0x1: Rewrite if DSCP is zero. This may be useful if a DSCP value of zero indicates that the host has not written any value to the DSCP field.
- 0x2: Rewrite selected DSCP values. In ANA::DSCP_CFG.DSCP_REWR_ENA specific DSCP values can be selected for rewrite, for example, if only certain DSCP values are allowed in the network.
- 0x3: Rewrite all DSCP values.

After a frame is selected for DSCP rewrite, based on the configuration for the ingress port, the new DSCP value is determined by mapping the QoS class and DP level to a new DSCP value (ANA::DSCP_REWR_CFG.DSCP_QOS_REWR_VAL).

This DSCP value is overruled by IS1 if a hit in IS1 returns an action vector with DSCP_ENA set.

The resulting DSCP value is forwarded to the Rewriter at the egress port, which determines whether to actually write the new DSCP value into the frame (REW::DSCP_CFG.DSCP_REWR_CFG). Optionally, the DSCP value may be translated before written into the frame (REW::DSCP_REMAP_CFG, REW::DSCP_REMAP_DP1_CFG) for applications where the switch acts as an DS boundary node.

When an IPv4 DSCP is rewritten, the IP header checksum is updated accordingly.

4.5.3 VOICE OVER IP (VOIP)

This section provides information about QoS in applications with Voice over IP (VoIP).

In a typical workgroup switch application with VoIP phones, both workstations and VoIP phones are connected to the switch. A workstation can be connected through a VoIP phone. Traffic from the workstation is usually untagged, whereas traffic from the VoIP phone may or may not be tagged. The QoS classification mechanism applied on the access port depends on the capabilities of the VoIP phone; these capabilities vary from phone to phone. With different VoIP phone models in the network, different access ports require different QoS classification mechanisms. The access switch can perform QoS classification, depending on the VoIP phone model, to achieve consistent VoIP QoS across the network.

Voice traffic can be identified in the following ways.

- **Source MAC address (OUI): Most vendors use a dedicated OUI for VoIP phones.**
- **EtherType**: Legacy phones may use a special EtherType for VoIP.
- **VID**: A special VID used for voice traffic.
- **UDP Port Range**: Voice traffic often uses a well-known port range for the Real-time Transport Protocol (RTP).
- **DSCP or ToS Precedence**: Many phones can set the DSCP value or the ToS precedence bits.
- **Priority Code Point**: Many phones send VLAN tagged frames and can set the priority code point.

All of these identification methods are supported by QoS classification through IS1. They can be used to determine the VoIP traffic's QoS class when entering the switch. For more information about the IS1, see Section 2.8.1.1, VCAP IS1 Port Configuration.

To ensure consistent QoS across the network, frames can be remarked on the uplink port. Priority Code Points and DSCP values can be remarked based on the QoS class determined by the QCLs. For more information about Priority Code Point and DSCP remarking, see Section 2.13.1, VLAN Editing, and Section 4.5.2, IPv4 and IPv6 DSCP Remarking.

Traffic received on the uplink port can usually rely on simple DSCP or PCP QoS classification.

4.6 VCAP Applications

This section provides information about Versatile Content Aware Processor (VCAP) applications for QoS classification, source IP guarding, and access control.

The following table shows the different control lists that the VCAP can be used to build.

TABLE 4-34: CONTROL LISTS AND APPLICATION

Control List	Description
Ingress control lists (ICLs)	QoS classification VLAN classification and translation policy association group classification
IPv4 source guarding control lists (S4CLs)	IPv4 source guarding
IPv6 source guarding control lists (S6CLs)	IPv6 source guarding
Access control lists (ACLs)	Access control
Egress control lists (ECLs)	Tagging and egress translations

4.6.1 NOTATION FOR CONTROL LISTS ENTRIES

Setting up a control list typically requires a large amount of register configurations. To maintain the overview of the VCAP functionality, the following control list notations are used. The register configurations are not listed. For more information about the VCAP configurations, see Section 2.8, VCAP.

The notation used is:

```
entry_number vcap entry_type {entry_field=value}
    → {action_field=value}
```

Each control entry in the notation consists of:

- The entry number specifying the TCAM address for the specific TCAM
- The VCAP used (IS1, IS2, ES0)
- The entry type (for instance NORMAL or MAC_ETYPE).
- Zero, one, or more entry fields with specified values. If no value is supplied, it is assumed that the value is 1.
- The action (indicated with →)
- Zero, one, or more action fields with specified values. If no value is supplied, it is assumed that the value is 1.

All entry fields not listed in the entry part of the control entry are set to don’t care.

All action fields not listed in the action part of the control entry are set to zero.

Default actions are special, because they do not have an entry type and a pattern to match:

```
default vcap (first|second) port=value
    →{action_field=value}
```

The notation is illustrated by the following examples.

Example 1:

An example of an ACL entry:

```
255 is2 ipv4_other first igr_port_mask=(1<<7) sip=10.10.12.134
    →
```

This ACL entry is located in entry number 255. It is matched for the first lookup, and it is part of the port ACL for port 7. The type is ipv4_other, and the action is not to change the normal flow for frames with `SIP = 10.10.12.134`.

Example 2:

Policy ACL A can include a monitoring rule that disables forwarding and learning of all incoming IPv4 traffic, but redirects a copy to CPU extraction queue number 3 using the hit-me-once filter. The hit-me-once filter enables the CPU to control when it ready to accept a new frame. The rule would look like this:

```
254 is2 ipv4_other first pag=A
    → hit_me_once cpu_qu_num = 3
```

Example 3:

This example shows an ACE that allows forwarding and learning of ARP requests from port 7, if the source IP address is `10.10.12.134`. The ACL entry also performs ARP sanity checks that frames must pass to match. The checks include checking that it is a Layer-2 broadcast, that the hardware address space is Ethernet, that the protocol address space is IP, that the MAC address and IP address lengths are correct, and that the sender hardware address (SMAC) matches the SMAC of the frame.

```
253 is2 arp first igr_port_mask=(1<<7) l2_bc opcode=arp_request
    sip=10.10.12.134
    arp_addr_space_ok arp_proto_space_ok arp_len_ok
    arp_sender_match
        →
```

Example 4:

If the default action from first lookup for port 7 is to discard all traffic, the following notation is used:

```
default is2 first port=7
    →mask_mode=1 port_mask=0x0
```

4.6.2 INGRESS CONTROL LISTS

The following table lists the registers associated with advanced QoS configuration through Ingress Control Lists.

TABLE 4-35: ADVANCED QOS CONFIGURATION REGISTER OVERVIEW

Register	Description	Replication
ANA:PORT:QOS_CFG	QoS configuration	Per port

Situation:

Assume a configuration with the following requirements:

- All frames with DSCP = 7 must get QoS class 2.
- All frames with TCP/UDP port numbers in the range 0 – 1023 must get QoS class 3, except frames with TCP/UDP port 25, which must get QoS class 1.
- All other frames must get QoS class 0.

Solution:

The resulting QoS Control List looks like this:

```
255 is1 normal first etype_len ip_snap dscp = 7
    → qos_ena=1, qos_val = 2
    254 is1 normal first etype_len ip_snap l4_sport = 25
        → qos_ena=1, qos_val = 1
    253 is1 normal first etype_len ip_snap etype = 25
        → qos_ena=1, qos_val = 1
    252 is1 normal first etype_len ip_snap l4_sport=(key:0, mask: 0x3FF)
        → qos_ena=1, qos_val = 3
    251 is1 normal first etype_len ip_snap etype = (key: 0, mask: 0x3FF)
        → qos_ena=1, qos_val = 3
ANA:PORT:QOS_CFG.QOS_DEFAULT_VAL = 0.
```

4.6.3 ACCESS CONTROL LISTS

The examples operate with three levels of ACLs:

- Port ACLs
- Policy ACLs
- Switch ACLs

The port ACLs are specific to a single port or a group of ports that form a link aggregation group. For example, a port ACL can be used for source IP filtering, locking a specific source IP address to a port. For more information about this example, see Section 4.6.4.1, Restrictive SIP Filter Using IS2.

The policy ACLs are shared for a group of ports that must have the same policy applied. For example, there could be one policy for ports through which workstations access the network and another policy for ports to which servers are connected.

The switch ACLs apply to all ports of the switch. They specify some general rules that apply to all traffic passing through the switch. The rules can still be rather specific, for example, covering a specific VLAN or a specific IP address.

In the examples, the resulting ACL can include one port ACL, one policy ACL, and the switch ACL. This is determined by the way the ingress port mask (IGR_PORT_MASK) and the policy association group (PAG) are used. For information about IGR_PORT_MASK and PAG, see Section 2.8.3, VCAP IS2. There are several ways to use the 8-bit PAG, but in this section, all eight bits are used to point out a policy ACL. The `IGR_PORT_MASK` points out the port ACL. This permits one port ACL per port and a total of 256 policy ACLs. Note that ports may share the same port ACL and a port by don't caring bits in the port ACL's IGR_PORT_MASK.

Each port has a default PAG assigned to it. The IS1 VCAP can be used to change the value of the PAG based on specific protocol fields matched in the IS1 lookup. The resulting PAG is used in the IS2 VCAP lookup and is matched against the PAG field of the ACL entries.

For an ACL entry in the IS2 VCAP, the PAG and IGR_PORT_MASK use this notation:

PAG = PolicyACL_ID
IGR_PORT_MASK = 1<<PortACL_ID

Note: The "<<" operator is the bitwise left shift operator. It shifts the left operand bit-wise to the left the number of positions specified by the right operand.

The IGR_PORT_MASK is a mask so the port number is left-shifted to create the mask.

For an ACL entry that is part of a port ACL for port 8, the PAG would be (*) and IGR_PORT_MASK would be (1<<8) = 0x100. The asterisk is a wildcard, which means that the PolicyACL_ID is a don't-care. For an ACL entry that is part of policy ACL A, the PAG would be (A) and the IGR_PORT_MASK would be (*). In this case, the PortACL_ID is a don't-care.

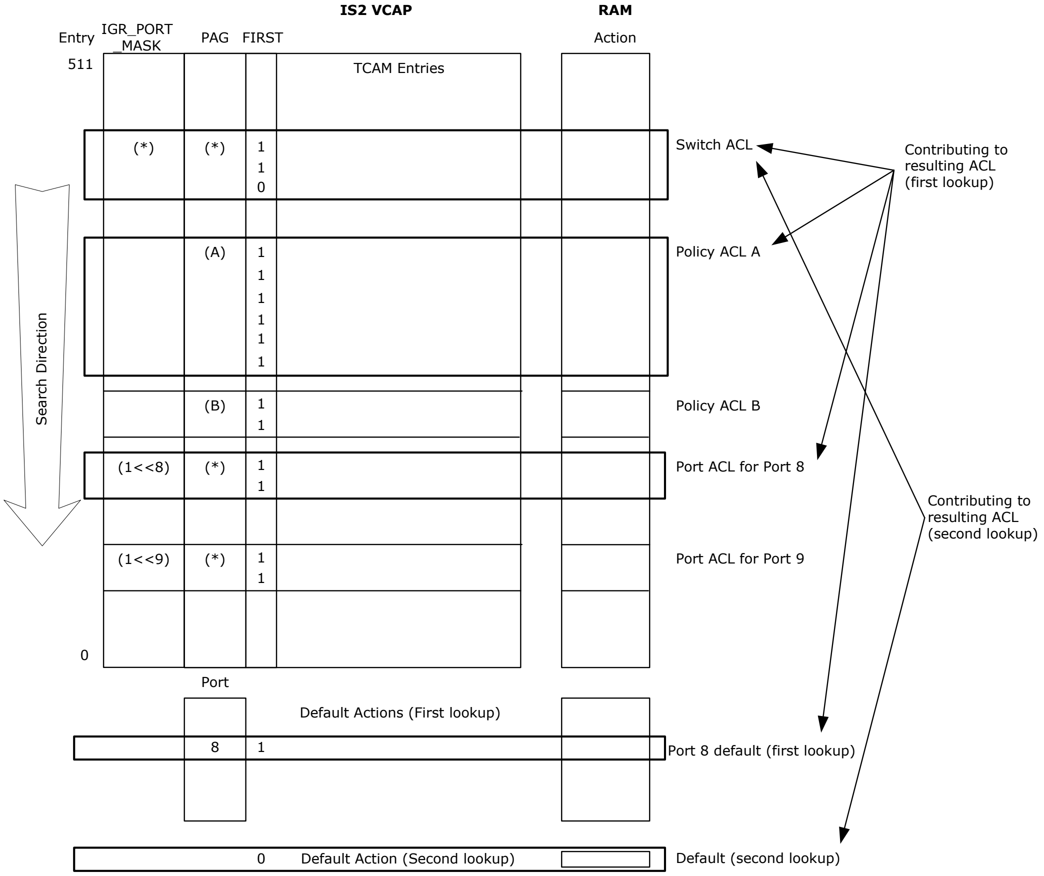
If, for example, port 8 must have policy A applied, the PAG assigned to port 8 is (A). Using this PAG value, the following ACLs match the lookup:

- Port ACL for port 8 with PAG = (*) and IGR_PORT_MASK = (1<<8)
- Policy ACL A with PAG = (A) and IGR_PORT_MASK = (*)
- Switch ACL with PAG = (*) and IGR_PORT_MASK = (*)

The ordering of the port ACL, the policy ACL, and switch ACL in the resulting ACL follows the ordering in the TCAM. In the following illustration, the switch ACL has the highest priority, followed by the policy ACL A, and finally, the port ACL for port 8.

The resulting ingress ACL in the example is made up of the ingress ACL entries in the switch ACL, the policy ACL A, the port ACL for port 8, and the default action for port 8. The VCAP also does a second lookup, for which the resulting ACL has a common default action as the last rule.

FIGURE 4-9: RESULTING ACL FOR LOOKUP WITH PAG = (A) AND IGR_PORT_MASK = (1<<8)



4.6.4 SOURCE IP FILTER (SIP FILTER)

The VCAP enables filtering of source IP (SIP) addresses on a port also known as source IP guarding. This can be used to only allow IP traffic from a specific SIP to enter the switch on a given port. Doing this can prevent the following denial of service (DoS) attacks: LAND attack, SMURF attack, SYN flood attack, Martian attack, and Ping attack.

4.6.4.1 Restrictive SIP Filter Using IS2

A restrictive SIP filter can be applied per port in networks where only IP traffic is allowed. The filter locks a specific SIP to the port and only permits ARP frames and IPv4 frames with the specified SIP to enter the switch on the given port.

For monitoring purposes, it is possible to permit IPv4 frames with other SIPs than the SIP locked to the port. The action is to redirect to the CPU, and the amount of traffic can be reduced by using the hit-me-once feature. The ACL entry for this can be part of a policy ACL for all ports on which the SIP filter is applied.

The port ACL has the following options:

- Permit IPv4 with trusted SIP
- Permit ARP with trusted SIP passing ARP sanity checks
- Permit all IPv4 — CPU redirect with hit-me-once filter (for monitoring)
- Default port action — discard all traffic

Situation:

Apply the restrictive SIP filter on port 7 with SIP `10.10.12.134`.

Resolution:

The resulting ACL for port 7 looks like this:

```
255 is2 ipv4_tcp_udp first igr_port_mask=(1<<7) sip=10.10.12.134
```

```
    →
254 is2 ipv4_other first igr_port_mask=(1<<7) sip=10.10.12.134
    →
253 is2 arp first igr_port_mask=(1<<7) l2_bc opcode=arp_request
    sip=10.10.12.134
    arp_addr_space_ok arp_proto_space_ok arp_len_ok
    arp_sender_match
    →
252 is2 ipv4_tcp_upd first pag=A
    → hit_me_once cpu_queue=3
251 is2 ipv4_other first pag=A
    → hit_me_once cpu_queue=3
default is2 first port=11
    →mask_mode=1 port_mask=0x0
```

Applying this SIP filter requires to two entries per port plus three common entries.

4.6.4.2 Restrictive SIP Filter Using IS1 and IS2

The same filter as listed above can be achieved using the host_match actions from IS1.

Situation:

Apply the restrictive SIP filter on port 7 with SIP `10.10.12.134`.

Resolution:

The resulting ACL for port 7 looks like this:

IS1:

```
255 is1 smac_sip4 igr_port=7 sip=10.10.12.134
    → host_match
```

IS2:

```
255 is2 ip4_tcp_upd first host_match=1
    →
254 is2 ip4_other first host_match=1
    →
253 is2 arp first igr_port_mask=(1<<7) l2_bc opcode=arp_request
    sip=10.10.12.134
    arp_addr_space_ok arp_proto_space_ok arp_len_ok
    arp_sender_match
    →
252 is2 ipv4_tcp_udp first pag=A
    → hit_me_once cpu_queue=3
251 is2 ipv4_other first pag=A
    → hit_me_once cpu_queue=3
default is2 first port=7
    →mask_mode=1 port_mask=0x0
```

Applying this SIP filter requires to one entry in IS1 per port and five common entries in IS2.

4.6.4.3 Less Restrictive SIP Filter Using IS2

For networks in which non-IP protocols are allowed, for example IPX and ARP, a less restrictive SIP filter can be applied with the following port ACL:

- Permit IPv4 with trusted SIP
- Discard all IPv4
- Default port action; Permit all traffic (non-IPv4, because all IPv4 traffic is covered by the ACL entries from other two items)

For monitoring purposes, the “Discard all IPv4” ACL can be changed to perform CPU redirect. This allows the CPU to monitor all incoming IPv4 frames with source IP addresses different from the trusted SIP, but without allowing these frames to be forwarded to other ports.

Situation:

Apply the less restrictive SIP filter on port 8 with source IP address 10.10.12.134, and monitor any IPv4 traffic with unauthorized source IP addresses with hit-me-once filtering to CPU extraction queue number 2. The monitoring rule is part of policy ACL A that is applied to all user ports.

Resolution:

The resulting ingress ACL for port 8 looks like this:

```
255 is2 ipv4_tcp_udp first igr_port_mask=(1<<8) sip=10.10.12.134
    →
254 is2 ipv4_other first igr_port_mask=(1<<8) sip=10.10.12.134
    →
63 is2 ipv4_tcp_udp first pag=A
    → hit_me_once cpu_queue=2
62 is2 ipv4_other first pag=A
    → hit_me_once cpu_queue=2
default is2 first port=10
    →
```

Applying this SIP filter requires two entries per port plus two common entries.

4.6.5 DHCP APPLICATION

A DHCP application can be supported using one policy ACL for the user ports and another policy ACL for the DHCP server ports.

On the user ports, the DHCP requests must be snooped to be able to automatically reset the SIP filters that are applied per port. DHCP replies should be prevented from being forwarded from user ports. For monitoring purposes, such illegal replies are redirected to the CPU.

On the DHCP server ports, DHCP replies are snooped to be able to automatically update the SIP filter for the user port where the reply goes.

In addition, an egress rule is needed to prevent forwarding of all DHCP requests to user ports.

Situation:

Policy ACL A is used for the user port DHCP policy, and policy ACL B is used for the DHCP server policy. The server ports are ports 8 and 9.

Snoop DHCP requests from user ports in CPU extraction queue 1, using policer 0 to protect the CPU. DHCP replies from the servers are snooped in queue 2, and are also subject to policing with policer 0. The illegal DHCP replies from user ports are redirected to queue 3 using the hit-me-once filter.

Resolution:

The PAG assigned to the user ports is (A). The PAG assigned to the DHCP server ports (8 and 9) is (B).

The following shows the ACL entries for the DHCP application:

```
255 is2 ipv4_tcp_udp protocol=udp
    sport=bootp_client dport=bootp_server
    → mask_mode=1 port_mask=0x0000300
63 is2 ipv4_tcp_udp first pag=A protocol=udp
    sport=bootp_client dport=bootp_server
    → cpu_copy_ena cpu_queue=1 police_ena police_idx=0
62 is2 ipv4_tcp_udp first pag=A protocol=udp
    sport=bootp_server dport=bootp_client
    → hit_me_once cpu_queue=3
31 is2 ipv4_tcp_udp first pag=B protocol=udp
    sport=bootp_server dport=bootp_client
    → cpu_copy_ena cpu_queue=2 police_ena police_idx=0
default is2 first
    → mask_mode=1 port_mask=0x0
default is2 second
    →
```

Regardless of the number of ports covered, four ACL entries are used: one in the switch ACL, two in policy ACL A, and one in policy ACL B.

4.6.6 ARP FILTERING

The VCAP support two useful ARP filters:

- Policing ARP requests to the switch’s IP address to mitigate DoS attacks by ARP flooding
- Performing general ARP sanity checks

Because these are general rules, it is sensible to make them part of the switch ACL.

Situation:

Discard all ARP frames that do not pass the ARP sanity checks. Police ARP requests to the switch’s IP address 10.10.12.1 using ACL policer 2. ACL policer 2 is configured to allow 16 frames per second, and the frames are copied to CPU extraction queue 0.

RARP is not allowed in the network.

Resolution:

To do ARP filtering in the switch ACL, perform the filtering for the switch’s IP address first, then allow all ARP frames passing the sanity checks, and finally, discard all remaining ARP frames. This is illustrated by the following:

```
        255 is2 arp first l2_bc opcode=arp_request
           dip=10.10.12.1
           arp_addr_space_ok arp_proto_space_ok arp_len_ok
           arp_sender_match
               → cpu_copy_ena cpu_queue=0 police_ena police_idx=255
    254 is2 arp first l2_bc opcode=(arp_request or arp_reply)
           arp_addr_space_ok arp_proto_space_ok arp_len_ok
           arp_sender_match
           →
        253 is2 arp
           →mask_mode=1 port_mask=0x0
```

The ACL policer configuration for policer 255 is done as follows:

```
# Set the base unit to 1 frame per second, enable the policer, and set the rate to 16 frames per
second and a burst of 1 frame:
ANA:POL[255]:POL_MODE_CFG.FRM_MODE = 1
ANA:POL[255]:POL_PIR_CFG.PIR_RATE = 16
ANA:POL[255]:POL_PIR_CFG.PIR_BURST = 3
```

Three ACL entries are used, irrespective of the number of ports covered.

4.6.7 PING POLICING

The network can easily be protected against ping attacks using a switch ACL rule that applies an ACL policer to all ping packets.

Situation:

Allow no more than 128 ping packets per second to be forwarded through the switch by means of ACL policer 15. Ping packets in excess of 128 frames per second are discarded.

Resolution:

Ping packets are ICMP frames with ICMP Type = Echo Request. Echo Request is specified by the first byte of the ICMP frame being 0x08. The rest of the ICMP frame is don’t-care. ICMP frames are carried in IPv4 frames with the protocol value 0x01.

The resulting switch ACL entry is as follows:

```
        127 is2 ipv4_other first protocol=icmp ip4_payload_high=0x8*
           → police_ena police_idx=15
```

ACL policer 15 in the policer pool is configured to 128 frames per second like this:

- ANA:POL[15]:POL_MODE_CFG.FRM_MODE = 1
- ANA:POL[15]:POL_PIR_CFG.PIR_RATE = 128
- ANA:POL[15]:POL_PIR_CFG.PIR_BURST = 1

One ACE is used, regardless of the number of ports covered.

4.6.8 TCP SYN POLICING

A server in the network can be protected against TCP SYN DoS attacks by policing TCP connection requests to the server’s IP address.

Situation:

Allow no more than 128 new TCP connections per second to the server with IP address 10.10.12.99. Use ACL policer 5.

Resolution:

TCP connection requests are TCP frames with the SYN flag set. The resulting switch ACL entry is as follows:

```
127 is2 ipv4_tcp_udp first protocol=tcp
    dip=10.10.12.99
    syn
    → police_ena police_idx=5
```

ACL policer 5 in the policer pool is configured to 128 frames per second by the following:

- ANA:POL[5]:POL_MODE_CFG.FRM_MODE = 1
- ANA:POL[5]:POL_PIR_CFG.PIR_RATE = 128
- ANA:POL[5]:POL_PIR_CFG.PIR_BURST = 1

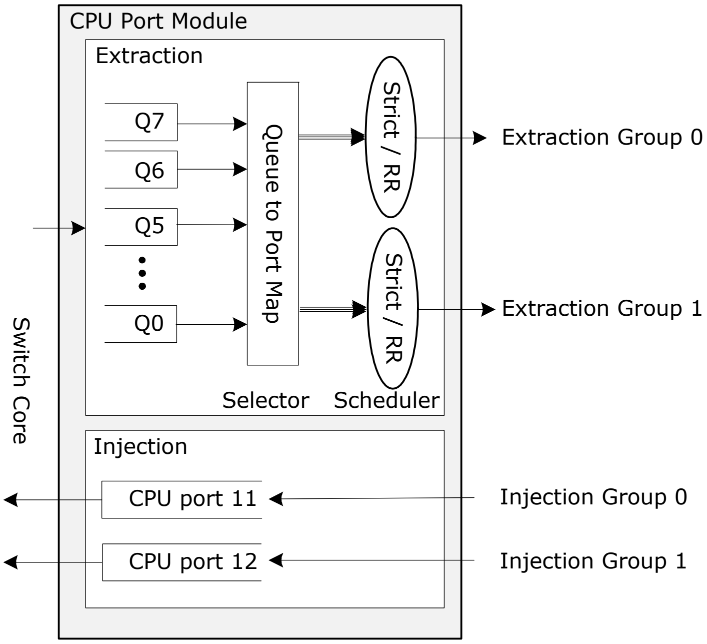
One ACE is used, regardless of the number of ports covered.

4.7 CPU Extraction and Injection

This section provides information about how the CPU extracts and injects frames to and from the switch core.

The following illustration shows the CPU port module used for injection and extraction.

FIGURE 4-10: CPU EXTRACTION AND INJECTION



The switch core forwards CPU extracted frames to eight CPU extraction queues. Each of these queue is then mapped to one of two CPU Extraction Groups. For each extraction group there is a scheduler (strict or round robin) which selects between the CPU extraction queues mapped to the same group.

When injecting frames, there are two CPU Injection Groups available where for instance one can be used for the Frame DMA and one can be used for manually injected frames. Both CPU injection groups have access to the switch core. However, within the switch-core, CPU injected frames are all seen as coming from port 11 in terms of queue system and analyzer configuration.

4.7.1 FORWARDING TO CPU

Several mechanisms can be used to trigger redirection or copying of frames to the CPU. They are listed in the following table.

TABLE 4-36: CONFIGURATIONS FOR REDIRECTING OR COPYING FRAMES TO THE CPU

Frame Type	Configuration (Including Selection of Extraction Queue)	Copy or Redirect
IEEE 802.1D Reserved Range DMAC = 01-80-C2-00-00-0x	ANA:PORT:CPU_FWD_BPDU_CFG ANA::CPUQ_8021_CFG.CPUQ_BPDU_VAL	Copy or redirect
IEEE 802.1D Allbridge DMAC = 01-80-C2-00-00-10	ANA:PORT:CPU_FWD_CFG.CPU_ALL-BRIDGE_REDIR_ENA ANA::CPUQ_CFG.CPUQ_ALLBRIDGE	Copy or redirect
IEEE 802.1D GARP Range DMAC = 01-80-C2-00-00-2x	ANA:PORT:CPU_FWD_GARP_CFG ANA::CPUQ_8021_CFG.CPUQ_GARP_VAL	Copy or redirect
IEEE 802.1D CCM/Link Trace Range DMAC = 01-80-C2-00-00-3x	ANA:PORT:CPU_FWD_CCM_CFG ANA::CPUQ_8021_CFG.CPUQ_CCM_VAL	Copy or redirect
IGMP (IPv4)	ANA:PORT: CPU_FWD_CFG.CPU_IGMP_REDIR_ENA ANA::CPUQ_CFG.CPUQ_IGMP	Redirect
IP Multicast Control (IPv4)	ANA:PORT: CPU_FWD_CFG.CPU_IPMC_CTRL_-COPY_ENA ANA::CPUQ_CFG.CPUQ_IPMC_CTRL	Copy
MLD (IPv6)	ANA:PORT: CPU_FWD_CFG.CPU_MLD_REDIR_ENA ANA::CPUQ_CFG.CPUQ_MLD	Redirect
Versatile Register Access Protocol (VRAP)	ANA:PORT: CPU_FWD_CFG.CPU_VRAP_REDIR_ENA ANA::CPUQ_CFG2.CPUQ_VRAP	Redirect
CPU-based learning	ANA:PORT:PORT_CFG.LEARNCPU ANA::CPUQ_CFG.CPUQ_LRN	Copy
CPU-based learning of locked MAC table entries seen on a new port	ANA:PORT: PORT_CFG.LOCKED_PORTMOVE_CPU ANA::CPUQ_CFG.CPUQ_LOCKED_PORT-MOVE	
CPU-based learning of frames exceeding learn limit in MAC table	ANA:PORT:PORT_CFG.LIMIT_CPU ANA::CPUQ_CFG.CPUQ_LRN	
MAC table match using MAC table	ANA::MACACCESS.MAC_CPU_COPY ANA::CPUQ_CFG.CPUQ_MAC_COPY	Copy
MAC table match using PGID table	ANA::MACACCESS.DEST_IDX ANA::PGID.PGID (bit 11) ANA::PGID.CPUQ_DST_PGID	Redirect or copy
Flooded frames	ANA::MACACCESS.DEST_IDX ANA::PGID.PGID (bit 11) ANA::PGID.CPUQ_DST_PGID	Redirect or copy
Any frame received on selected ports	ANA:PORT:CPU_SRC_COPY_ENA ANA:CPUQ_CFG.CPUQ_SRC_COPY	Copy
Mirroring	ANA::MIRRORPORTS (bit 11) ANA::CPUQ_CFG:CPUQ_MIRROR For more information about mirroring, see Section 4.3.10, Mirroring.	Copy
VCAP IS2 rules	For more information about IS2, see Section 2.8.3, VCAP IS2.	Redirect or copy

TABLE 4-36: CONFIGURATIONS FOR REDIRECTING OR COPYING FRAMES TO THE CPU

Frame Type	Configuration (Including Selection of Extraction Queue)	Copy or Redirect
SFlow	ANA::CPUQ_CFG.CPUQ_SFLOW For more information about SFlow, see Section 2.9.3.7, sFlow Sampling.	Copy

4.7.2 FRAME EXTRACTION

The CPU receives frames through the eight CPU extraction queues in the CPU port module. The eight queues are using resources (memory and frame descriptor pointers) from the shared queue system and are subject to the thresholds and congestion rules programmed for the CPU port (port 11) and the shared queue system in general.

The CPU can read frames from the CPU extraction queues in two ways:

- Reading registers in the CPU port module. For more information, see Section 2.14.1, Frame Extraction.
- FDMA from CPU port module to RAM. For more information, see Section 3.7, Frame DMA.

In addition, the VRAP engine may attach to one of the CPU extraction groups and use this to receive and process all extracted VRAP requests.

The switch core may place the 20-byte long CPU extraction header before the DMAC or after the SMAC (SYS::PORT_-MODE.INCL_XTR_HDR). The CPU extraction header contains relevant side band information about the frame such as the frame's classification result (VLAN tag information, DSCP, QoS class, Rx time stamp) and the reason for sending the frame to the CPU. For more information about the contents of the CPU extraction header, see Table 2-113.

4.7.3 FRAME INJECTION

The CPU can inject frames through the two CPU injection groups. The injection queues use resources (memory and frame descriptor pointers) from the shared queue system and are subject to the thresholds and congestion rules programmed for the CPU port (port 11) and the shared queue system in general.

The CPU can write frames to the CPU injection groups in two ways:

- Registers access to the CPU port module. For more information, see Section 4.7, CPU Extraction and Injection.
- FDMA to CPU port module. For more information, see Section 3.7, Frame DMA.

In addition, the VRAP engine may attach to one of the CPU injection groups and transmit and use this for transmitting VRAP replies.

The first 20 bytes of a frame written into a CPU group is an injection header containing relevant side band information about how the frame must be processed by the switch core (SYS::PORT_MODE.INCL_INJ_HDR). For more information, see Table 2-94.

4.7.4 FRAME EXTRACTION AND INJECTION USING AN EXTERNAL CPU

The following table lists the configuration registers associated with using an external CPU.

TABLE 4-37: CONFIGURATION REGISTERS WHEN USING AN EXTERNAL CPU

Register/Register Field	Description / Value	Replication
QSYS::EXT_CPU_CFG.EXT_CPU_PORT	Port number where external CPU is connected.	None
QSYS::EXT_CPU_CFG.EXT_CPUQ_MSK	Configures which CPU extraction queues are sent to the external CPU.	None
SYS::PORT_MODE.INCL_XTR_HDR	Enables the insertion of the CPU extraction header in egress frames.	Per port
SYS::PORT_MODE.INCL_INJ_HDR	Enables ingress port to look for CPU injection header in incoming frames.	Per port

An external CPU can connect up to any front port module and use the Ethernet interface for extracting and injecting frames into the switch core.

Note: If an external CPU is connected by means of the serial interface or PCIe interface, the frame extraction and injection is performed. For more information, see Section 4.7.2, Frame Extraction and Section 4.7.3, Frame Injection.

When injecting or extracting frames, the CPU injection or extraction header is placed before the DMAC with an optional prefix. For more information about the prefixing, see Section 2.14.3, Node Processor Interface (NPI). When injecting frames, the CPU injection header controls whether a frame is processed by the analyzer or forwarded directly to the destination set specified in the injection header.

An internal and external CPU may coexist in a dual CPU system where the two CPUs handles different run-time protocols. When extracting CPU frames, it is selectable which CPU extraction queues are connected to the external CPU and which remain connected to the internal CPU (SYS::EXT_CPU_CFG.EXT_CPUQ_MSK). If a frame is forwarded to the CPU for more than one reason (for example, a BPDU which is also a learn frame), the frame can be forwarded to both the internal CPU extraction queues and to the external CPU.

5.0 REGISTERS

Information about the registers for this product is available in the attached file. To view or print the information, double-click the attachment icon.

6.0 ELECTRICAL SPECIFICATIONS

This section provides the DC characteristics, AC characteristics, recommended operating conditions, and stress ratings for the VSC7513 device.

6.1 DC Specifications

This section contains the DC characteristics for the VSC7513 device.

6.1.1 INTERNAL PULL-UP OR PULL-DOWN RESISTORS

The following table lists the specifications for the internal resistors used by I/O signals in the device. For more information, see Section 7.2, Pins by Function.

TABLE 6-1: INTERNAL RESISTOR CHARACTERISTICS

Parameter	Symbol	Minimum	Maximum	Unit
Internal Pull-up resistor	R_{PU}	25	90	kΩ
Internal Pull-down resistor	R_{PD}	25	112	kΩ

6.1.2 REFERENCE CLOCK INPUTS

The following table lists the reference clock input specifications.

TABLE 6-2: REFERENCE CLOCK INPUT CHARACTERISTICS

Parameter	Symbol	Minimum	Typical	Maximum	Unit
Input voltage range	V_{IP}, V_{IN}	–25	—	1200	mV
Single-ended input swing	$\|V_{SE}\|$	600	—	1000[1]	mV_{pp}
Differential peak-to-peak input swing	$\|V_{ID}\|$	200	—	1200	mV_{ppd}
Input common-mode voltage	V_{CM}		2/3 x V_{DD}	—	mV

[1] Input common-mode voltage and amplitude must not exceed 1200 mV.

6.1.3 PLL CLOCK OUTPUTS

The following table lists the PLL clock outputs specifications.

TABLE 6-3: PLL CLOCK OUTPUTS CHARACTERISTICS

Parameter	Symbo l	Minimum	Typical	Maximum	Unit	Condition
Differential resistance	R_{DIFF}	80	100	120	Ω	
Differential peak-to-peak output swing[1]	V_{OD}	290	—	510	mV_{ppd}	V_{DD_A} = 1.0V
Differential peak-to-peak output swing[1]	V_{OD}	360	—	635	mV_{ppd}	V_{DD_A} = 1.0V[2]
Output common-mode voltage	V_{CM}	V_{DD_A} – 500 mV	—	V_{DD_A}	mV	—

[1] Output swing is register programmable in 16 steps.

[2] Driver amplitude depends on driver-to-receiver adaptation configured in register. Increased amplitudes can be achieved if receiver is common-mode terminated to V_{DD_A}, and if driver is configured accordingly.

6.1.4 DDR3 SDRAM SIGNALS

The following table lists the specifications of DDR3 SDRAM signals.

TABLE 6-4: DDR3 SDRAM SIGNALS CHARACTERISTICS

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Input reference voltage[1]	DDR_V_{REF}	49% V_{DD_IODDR}	51% V_{DD_IODDR}	V	V_{DD_IODDR} = 1.5V
Input voltage high	$V_{IH(DC)}$	DDR_V_{REF} + 0.100	V_{DD_IODDR}	V	—
Input voltage low	$V_{IL(DC)}$	–0.3	DDR_V_{REF} – 0.100	V	—
Output voltage high	$V_{OH(DC)}$	0.8 x V_{DD_IODDR}	—	V	Not terminated, 1 pF load
Output voltage low	$V_{OL(DC)}$	–0.3	0.2 x V_{DD_IODDR}	V	Not terminated, 1 pF load
Input leakage current	$\|I_L\|$	—	40	µA	0 V ≤ V_I ≤ V_{DD_IODDR}
Output source DC current[2]	I_{OH}	–6	—	mA	External 40 Ω termination to V_{DD_IODDR}/2
Output sink DC current[2]	I_{OL}	6	—	mA	External 40Ω termination to V_{DD_IODDR}/2

[1] DDR_V_{REF} is expected to track variations in V_{DD_IODDR}. Peak-to-peak AC noise on DDR_V_{REF} must not exceed ±2% of DDR_$V_{REF.}$

[2] With 34Ω output driver impedance.

6.1.5 DDR3L SDRAM SIGNALS

The following table lists the specifications of DDR3L SDRAM signals.

TABLE 6-5: DDR3L SDRAM SIGNALS CHARACTERISTICS

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Input reference voltage[1]	DDR_V_{REF}	49% V_{DD_IODDR}	51% V_{DD_IODDR}	V	V_{DD_IODDR} = 1.5V
Input voltage high	$V_{IH(DC)}$	DDR_V_{REF} + 0.09	V_{DD_IODDR}	V	—
Input voltage low	$V_{IL(DC)}$	–0.3	DDR_V_{REF} – 0.09	V	—
Output voltage high	$V_{OH(DC)}$	0.8 x V_{DD_IODDR}		V	Not terminated, 1 pF load
Output voltage low	$V_{OL(DC)}$	—	0.2 x V_{DD_IODDR}	V	Not terminated, 1 pF load
Input leakage current	$\|I_L\|$	—	40	µA	0 V ≤ V_I ≤ V_{DD_IODDR}
Output source DC current[2]	I_{OH}	–6	—	mA	External 40 Ω termination to V_{DD_IODDR}/2
Output sink DC current[2]	I_{OL}	6	—	mA	External 40 Ω termination to V_{DD_IODDR}/2

[1] DDR_V_{REF} is expected to track variations in V_{DD_IODDR}. Peak-to-peak AC noise on DDR_V_{REF} must not exceed ±2% of DDR_V_{REF}.

[2] With 34 Ω output driver impedance.

6.1.6 SERDES1G

The following table lists the SERDES1G specifications.

TABLE 6-6: SERDES1G CHARACTERISTICS FOR 1G TRANSMITTER

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Condition
Differential resistance	R_{DIFF}	80	100	120	Ω	—
Output voltage high	V_{OH}		—	1050	mV	—
Output voltage low	V_{OL}	0	—		mV	—
Differential peak-to-peak output voltage[1]	V_{O_DIFF}	300	—	800	mV_{ppd}	100BASE-FX, SGMII
Differential peak-to-peak output voltage[1]	V_{O_DIFF}	500	—	1200	mV_{ppd}	SFP
Differential peak-to-peak output voltage[1]	V_{O_DIFF}	800	—	1100	mV_{ppd}	1000BASE-KX
Differential peak-to-peak output voltage with Tx disabled	V_{O_IDLE}	—	—	30	mV_{ppd}	—

1. Output amplitude is configurable in 16 steps.

TABLE 6-7: SERDES1G CHARACTERISTICS FOR 1G RECEIVER

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Conditions
Differential resistance	R_{DIFF}	80	100	120	Ω	—
Absolute input voltage range	V_{IN}	–25	—	1200	mV	—
Common-mode voltage	V_{CM_AC}	0	—	V_{DD_A}	mV	AC-coupled operation[1]
Common-mode voltage	V_{CM_DC}	—	0.7 x V_{DD_A}		mV	DC-coupled operation[2, 3]
Differential peak-to-peak input voltage	V_{IN_DIFF}	100		1600	mV_{ppd}	See note[4]

1. Compatibility to SGMII transmitters requires external AC-coupling. The maximum common-mode voltage is provided without a differential signal. It is limited by the minimum and maximum input voltage range and the signal amplitude of input.
2. For more information about optional DC-coupling, contact your Microchip sales representative.
3. Common-mode termination disabled. The maximum differential peal-to-peak input is limited by the maximum input voltage range.
4. Applies to all the supported modes. For 100BASE-FX mode, disable internal AC-coupling.

6.1.7 SERDES6G

The following table lists the SERDES6G specifications.

TABLE 6-8: SERDES6G CHARACTERISTICS FOR 6G TRANSMITTER

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Condition
Differential resistance	R_{DIFF}	80	100	120	Ω	—
Output voltage high	V_{OH}	—	—	V_{DD_VS}	mV	—
Output voltage low	V_{OL}	0	—		mV	—
Differential peak-to-peak output voltage[1]	V_{O_DIFF}	300	—	800	mV_{ppd}	100BASE-FX, SGMII[2]
Differential peak-to-peak output voltage[1]	V_{O_DIFF}	400	—	750	mV_{ppd}	QSGMII ob_post0 = 6 ob_lv = 39
Differential peak-to-peak output voltage[1]	V_{O_DIFF}	500	—	1200	mV_{ppd}	SFP, 2.5G

TABLE 6-8: SERDES6G CHARACTERISTICS (CONTINUED) FOR 6G TRANSMITTER

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Condition
Differential peak-to-peak output voltage[1]	V_{O_DIFF}	800	—	1200	mV_{ppd}	1000BASE-KX and PCIe[3]
Differential peak-to-peak output voltage with Tx disabled	V_{O_IDLE}	—	—	30	mV_{ppd}	—
Output current, driver shorted to GND	T_ISG	—	—	40	mA	—

1. Drive level depends on register configuration.
2. Compatibility to SGMII receiver requires AC-coupling.
3. Compatibility to supported standards requires 1.2 V supply for driver.

TABLE 6-9: SERDES6G CHARACTERISTICS FOR 6G RECEIVER

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Condition
Differential resistance	R_{DIFF}	80	100	120	Ω	—
Absolute input voltage range	V_{IN}	–25		1200	mV	—
Common-mode voltage	V_{CM}	0	Internal V_{CM}	—	mV	AC-coupled operation[1]
Common-mode voltage	V_{CM}	—	V_{DD_A}	—	mV	DC-coupled operation, load type 2[1]
Differential peak-to-peak input voltage	V_{IN_DIFF}	100	—	1600	mV_{ppd}	See note[2]

1. Mode for common-mode termination is specified by configuration register setting. Input amplitude in DC-coupled mode must not exceed maximum input voltage range. For more information about optional DC-coupling, contact your Microchip representative.
2. Compatibility to SGMII transmitter requires AC-coupling.

6.1.8 GPIO, SI, JTAG, AND MISCELLANEOUS SIGNALS

The following table lists the specifications of GPIO, SI, JTAG, and miscellaneous signals.

TABLE 6-10: GPIO, SI, JTAG, AND MISCELLANEOUS SIGNALS CHARACTERISTICS

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Condition
Output high voltage[1]	V_{OH}	2.1	2.35	—	V	I_{OH} = –2 mA
Output high voltage[1]	V_{OH}	1.7	2.0	—	V	I_{OH} = –12 mA
Output low voltage	V_{OL}	—	—	0.4	V	I_{OL} = 2 mA
Output low voltage	V_{OL}	—	—	0.7	V	I_{OL} = 12 mA
Input high voltage	V_{IH}	1.85	—	3.6	V	—
Input low voltage	V_{IL}	–0.3	—	0.8	V	—
Input high current[2]	I_{IH}	—	—	10	µA	V_I = VDD_IO
Input low current[2]	I_{IL}	–10	—		µA	V_I = 0V
Input capacitance	C_I	—	—	10	pF	—

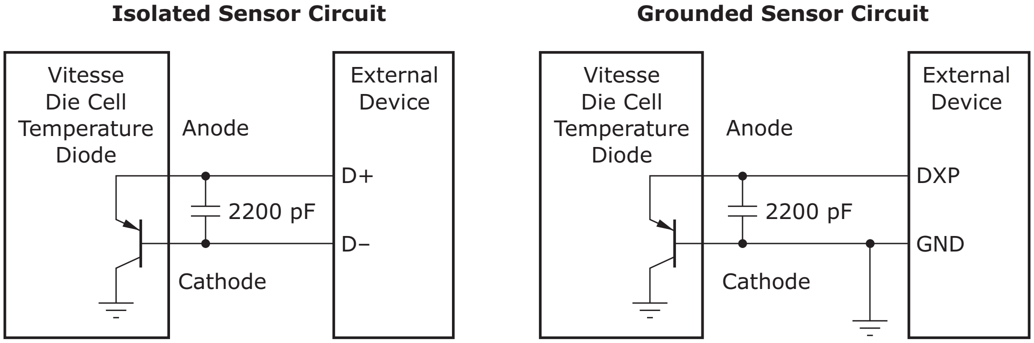
1. VDD_IO=2.38V for minimum, VDD_IO=2.50V for typical.
2. Input high current and input low current equals the maximum leakage current, excluding the current in the built-in pull resistors.

6.1.9 THERMAL DIODE

The device includes an on-die diode and internal circuitry for monitoring die temperature (junction temperature). The operation and accuracy of the diode is not guaranteed and should only be used as a reference.

The on-die thermal diode requires an external thermal sensor, located on the board or in a stand-alone measurement kit. Temperature measurement using a thermal diode is very sensitive to noise. The following illustration shows a generic application design.

FIGURE 6-1: THERMAL DIODE



Note: Microchip does not support or recommend operation of the thermal diode under reverse bias.

The following table provides the diode parameter and interface specifications with the pins connected internally to VSS in the device.

TABLE 6-11: THERMAL DIODE PARAMETERS

Parameter	Symbol	Typical	Maximum	Unit
Forward bias current	IFW	See note[1]	1	mA
Diode ideality factor	n	1.008	—	—

[1] Typical value is device dependent.

The ideality factor, n, represents the deviation from ideal diode behavior as exemplified by the following diode equation:

$$I_{FW} = I_S(e^{(qV_D)/(nkT)} - 1)$$

where, I_S = saturation current, q = electron charge, V_D = voltage across the diode, k = Boltzmann constant, and T = absolute temperature (Kelvin).

6.2 AC Specifications

This section contains the AC characteristics for the VSC7513 device.

6.2.1 REFCLK REFERENCE CLOCK (1G AND 6G SERDES)

The following table lists the REFCLK reference clock input specifications.

TABLE 6-12: REFERENCE CLOCK INPUT CHARACTERISTICS

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Condition
REFCLK frequency REFCLK_CONF = 100	—	–100 ppm	25	100 ppm	MHz	—
REFCLK frequency REFCLK_CONF = 000	—	–100 ppm	125	100 ppm	MHz	—
REFCLK frequency REFCLK_CONF = 001	—	–100 ppm	156.25	100 ppm	MHz	—
REFCLK frequency REFCLK_CONF = 010	—	–100 ppm	250	100 ppm	MHz	—
Clock duty cycle	—	40	—	60	%	Measured at 50% threshold
Rise time and fall time	t_R, t_F	—	—	1.5	ns	Within +/– 200 mV relative to V_{DD} x 2/3
Jitter transfer from REFCLK to SERDES, bandwidth from 10 kHz to 100 kHz	—	—	—	0.3	dB	REFCLK = 25 MHz
Jitter transfer from REFCLK to SERDES, bandwidth from 100 kHz to 4 MHz	—	—	—	1	dB	REFCLK = 25 MHz

TABLE 6-12: REFERENCE CLOCK INPUT CHARACTERISTICS (CONTINUED)

Parameter	Symbo l	Minimum	Typical	Maximum	Unit	Condition
Jitter transfer from REFCLK to SERDES, bandwidth above 4 MHz	—	—	—	1 – 20 × log (ƒ/4 MHz)	dB	REFCLK = 25 MHz
Jitter transfer from REFCLK to SERDES, bandwidth from 10 kHz to 300 kHz	—	—	—	0.3	dB	REFCLK ≥ 125 MHz
Jitter transfer from REFCLK to SERDES, bandwidth from 300 kHz to 3 MHz	—	—	—	0.6	dB	REFCLK ≥ 125 MHz
Jitter transfer from REFCLK to SERDES , bandwidth from 3 MHz to 12 MHz	—	—	—	2	dB	REFCLK ≥ 125 MHz
Jitter transfer from REFCLK to SERDES , bandwidth above 12 MHz	—	—	—	2 – 20 × log (ƒ/12 MHz)	dB	REFCLK ≥ 125 MHz
REFCLK input peak-to-peak jitter, bandwidth from 2.5 kHz and 10 MHz[1]	—	—	—	20	ps	To meet G.8262 1G SyncE jitter generation specification

[1] Peak-to-peak values are typically higher than the RMS value by a factor of 10 to 14.

6.2.2 PLL CLOCK OUTPUTS

The following table lists the PLL clock outputs specifications.

TABLE 6-13: PLL CLOCK OUTPUTS CHARACTERISTICS

Parameter	Symbo l	Minimu m	Maximu m	Unit	Condition
Data rate	—		625	MHz	—
Clock duty cycle	t_C	45	55	%	Measured at 50% threshold
Rise time and fall time	t_R, t_F	100	300	ps	20% to 80% of V_S
Intrapair skew	t_{SKEW}	—	100	ps	—
Jitter generation, 10 kHz to 50 MHz	—	—	4	ps_{RMS}	Jitter-free input used for REFCLK

6.2.3 SERDES1G

The following table lists the SERDES1G specifications (applies to the S[5:0]_TXN/P pins).

TABLE 6-14: SERDES1G CHARACTERISTICS FOR 100BASE-FX, SGMII, SFP, 1000BASE-KX TRANSMITTER

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Data rate	—	125 – 100 ppm	125 + 100 ppm	Mbps	100BASE-FX
Data rate	—	1.25 – 100 ppm	1.25 + 100 ppm	Gbps	SGMII, SFP, 1000BASE-KX
Differential output return loss	RLO_{SDD22}	—	– 10	dB	50 MHz to 625 MHz
Differential output return loss	RLO_{SDD22}	—	–10 + 10 x log(f/ 625 MHz)	dB	625 MHz to 1250 MHz
Rise time and fall time[1]	t_R, t_F	60	300	ps	20% to 80%
Interpair skew	t_{SKEW}	—	20	ps	

TABLE 6-14: SERDES1G CHARACTERISTICS FOR 100BASE-FX, SGMII, SFP, 1000BASE-KX TRANSMITTER (CONTINUED)

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Deterministic jitter	DJ	—	80	ps	Measured according to IEEE 802.3 Clause 38.5
Total jitter	TJ	—	192	ps	Measured according to IEEE 802.3 Clause 38.5
Wideband SyncE jitter	WJT	—	0.5	UI_{P-P}	Measured according to ITU-T G.8262 section 8.3

[1] Slew rate is programmable.

TABLE 6-15: SERDES1G CHARACTERISTICS FOR 100BASE-FX, SGMII, SFP, 1000BASE-KX RECEIVER

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Data rate		125 – 100 ppm	125 + 100 ppm	Mbps	100BASE-FX
Data rate		1.25 – 100 ppm	1.25 + 100 ppm	Gbps	SGMII, SFP, 1000BASE-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 1250 MHz
Jitter tolerance, total[1]	TOL_{TJ}	600	—	ps	SGMII, SFP, 1000BASE-KX. Measured according to IEEE 802.3 Clause 38.6.8
Jitter tolerance, deterministic[1]	TOL_{DJ}	370	—	ps	SGMII, SFP, 1000BASE-KX. Measured according to IEEE 802.3 Clause 38.6.8
Jitter tolerance, duty cycle distortion	TOL_{DCD}	1.4	—	ns, p-p	100BASE-FX. Measured according to ISO/IEC 9314-3:1990
Jitter tolerance, data dependent	TOL_{DDJ}	2.2	—	ns, p-p	100BASE-FX. Measured according to ISO/IEC 9314-3:1990
Jitter tolerance, random	TOL_{RJ}	2.27	—	ns, p-p	100BASE-FX. Measured according to ISO/IEC 9314-3:1990
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 6-15: SERDES1G CHARACTERISTICS (CONTINUED) FOR 100BASE-FX, SGMII, SFP, 1000BASE-KX RECEIVER (CONTINUED)

Parameter	Symbol	Minimum	Maximum	Unit	Condition
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

[1] Jitter requirements represent high-frequency jitter (above 637 kHz) and not low-frequency jitter or wander.

6.2.4 SERDES6G

The following table lists the SERDES6G specifications (applies to the S[8:6]_TXN/P pins).

TABLE 6-16: SERDES6G CHARACTERISTICS FOR 100BASE-FX, SGMII, SFP, 2.5G, 1000BASE-KX TRANSMITTER

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Data rate	—	125 – 100 ppm	125 + 100 ppm	Mbps	100BASE-FX
Data rate	—	1.25 – 100 ppm	125 + 100 ppm	Gbps	SGMII, SFP, 1000BASE-KX
Data rate	—	3.125 – 100 ppm	3.125 + 100 ppm	Gbps	2.5G
Differential output return loss	RLO_{SDD22}	—	–10	dB	50 MHz to 625 MHz
Differential output return loss	RLO_{SDD22}	—	–10 + 10 x log(f/625 MHz)	dB	625 GHz to 1250 MHz
Rise time and fall time[1]	t_R, t_F	60	320	ps	20% to 80%
Interpair skew	t_{SKEW}	—	20	ps	—
Random jitter	RJ	—	0.15	UI_{P-P}	At BER 10^{-12}
Deterministic jitter	DJ	—	0.10	UI_{P-P}	—
Total jitter	TJ	—	0.25	UI_{P-P}	—
Wideband SyncE jitter	TWJ	—	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[2]	—	mV	—
Eye mask	Y2		800	mV	—

[1] Slew rate is programmable.
[2] Compatibility to supported standards requires 1.2V supply for driver.

TABLE 6-17: SERDES6G CHARACTERISTICS FOR 100BASE-FX, SGMII, SFP, 2.5G, 1000BASE-KX RECEIVER

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Data rate	—	125 – 100 ppm	125 + 100 ppm	Mbps	100BASE-FX
Data rate	—	1.25 – 100 ppm	1.25 + 100 ppm	Gbps	SGMII, SFP, 1000BASE-KX
Data rate	—	3.125 – 100 ppm	3.125 + 100	Gbps	2.5G
Differential input return loss	RLI_{SDD11}	—	–10	dB	50 MHz to 625 MHz

TABLE 6-17: SERDES6G CHARACTERISTICS FOR 100BASE-FX, SGMII, SFP, 2.5G, 1000BASE-KX RECEIVER (CONTINUED)

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Differential input return loss	RLI_{SDD11}		–10 + 10 x log(f/65 MHz)2	dB	625 MHz to 1250 MHz
Jitter tolerance, total[1]	TOL_{TJ}	600	—	ps	Measured according to IEEE 802.3 Clause 38.6.8
Jitter tolerance, deterministic[1]	TOL_{DJ}	370	—	ps	Measured according to IEEE 802.3 Clause 38.6.8
Jitter tolerance, duty cycle distortion	TOL_{DCD}	1.4	—	ns, p-p	100BASE-FX. Measured according to ISO/IEC 9314-3:1990
Jitter tolerance, data dependent	TOL_{DDJ}	2.2	—	ns, p-p	100BASE-FX. Measured according to ISO/IEC 9314-3:1990
Jitter tolerance, random	TOL_{RJ}	2.27	—	ns, p-p	100BASE-FX. Measured according to ISO/IEC 9314-3:1990
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
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

[1] Jitter requirements represent high-frequency jitter (above 637kHz) and not low-frequency jitter or wander.

TABLE 6-18: SERDES6G CHARACTERISTICS FOR QSGMII TRANSMITTER

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Data rate	—	5.0 – 100 ppm	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[1]	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}	—

TABLE 6-18: SERDES6G CHARACTERISTICS FOR QSGMII TRANSMITTER (CONTINUED)

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Duty cycle distortion (part of DJ)	DCD	—	0.05	UI_{P-P}	—
Total jitter	TJ	—	0.30	UI_{P-P}	—
Eye mask	X1	—	0.15	UI_{P-P}	Near-end
Eye mask	X2	—	0.40	UI_{P-P}	Near-end
Eye mask	Y1	200	—	mV	Near-end
Eye mask	Y2	—	450	mV	Near-end

1. Slew rate is programmable.

TABLE 6-19: SERDES6G CHARACTERISTICS FOR QSGMII RECEIVER

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Data rate		5.0 – 100 ppm	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
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	—	0.30	UI_{P-P}	—
Total jitter	TJ	—	0.60	UI_{P-P}	Sinusoidal jitter excluded
Eye mask	R_X1	—	0.30	UI_{P-P}	—
Eye mask	R_Y1	50		mV	—
Eye mask	R_Y2	—	450	mV	—

TABLE 6-20: SERDES6G CHARACTERISTICS FOR PCIE TRANSMITTER

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Condition
Data rate	—	2.5 – 300 ppm		2.5 + 300 ppm	Gbps	—
Differential output return loss	RLO_{SDD22}	—		–10	dB	50 MHz to 1.25 GHz
Common-mode return loss	RLO_{SCC22}	—		–6	dB	50 MHz to 1.25 GHz
Demphasized differential output voltage (ratio)	Tode	–4	–3.5	–3	dB	Register setting post0 = 18
Rise time and fall time[1]	t_R, t_F	60		130	ps	20% to 80%. Recommended value.

TABLE 6-20: SERDES6G CHARACTERISTICS FOR PCIE TRANSMITTER (CONTINUED)

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Condition
Total jitter	TJ	—	—	0.25	UI_{P-P}	Near-end
Differential amplitude	$V_{TX\_DIFF\_PP}$	800[2]	—	1200	mV_{P-P}	Near-end

[1] Slew rate is programmable. Configure accordingly for compliance to supported standard.
[2] Compatibility to supported standards requires 1.2V supply for driver.

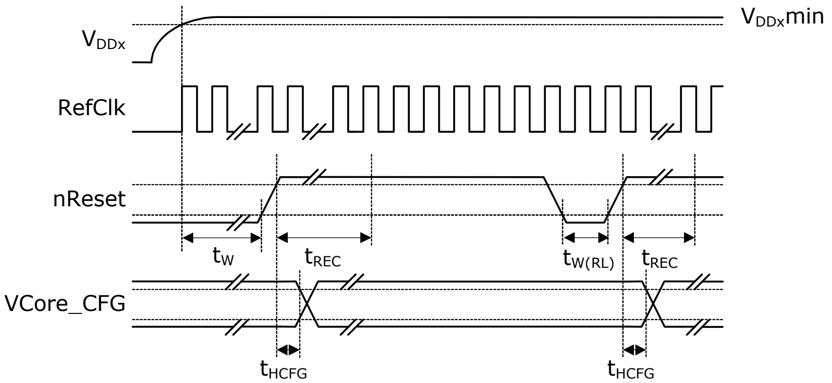
TABLE 6-21: SERDES6G CHARACTERISTICS FOR PCIE RECEIVER

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Data rate	—	2.5 – 300 ppm	2.5 + 300 ppm	Gbps	—
Differential input return loss	RLI_{SDD11}	—	–10	dB	50 MHz to 1.25 GHz
Common-mode return loss	RLI_{S-CC11}	—	–6	dB	80 MHz to 2.25 GHz
Total jitter tolerance	TJ	—	0.60	UI_{P-P}	—
Eye mask	R_X1	—	0.20	UI	—
Eye mask	R_X2	—	0.50	UI	—
Eye mask	R_Y1	85		mV	—
Eye mask	R_Y2	—	600	mV	—

6.2.5 RESET TIMING SPECIFICATIONS

The following illustration shows the nReset signal waveform and the required measurement points for the timing specification.

FIGURE 6-2: RESET SIGNAL TIMING



The following table lists the specifications for the signal applied to the nReset input at the reset pin.

TABLE 6-22: RESET TIMING CHARACTERISTICS

Parameter	Symbol	Minimum	Maximum	Unit
nReset assertion time after power supplies and clock stabilizes	t_W	2	—	ms
Recovery time from reset inactive to device fully active	t_{REC}	—	10	ms
nReset pulse width	$t_{W(RL)}$	100	—	ns

TABLE 6-22: RESET TIMING CHARACTERISTICS

Parameter	Symbol	Minimum	Maximum	Unit
Hold time for GPIO-mapped strapping pins relative to nRESET	t_{HCFG}	50	—	ns

6.2.6 MIIM TIMING SPECIFICATIONS

The following table lists the specifications for the signal applied to the MIM input at the MIM pin.

TABLE 6-23: MIIM TIMING CHARACTERISTICS

Parameter	Symbol	Minimum	Maximum	Unit	Condition
MDC frequency[1]	f	0.448	20.83	MHz	—
MDC cycle time[2]	t_C	48	2048	ns	—
MDC time high	$t_{W(CH)}$	20	—	ns	C_L = 50 pF
MDC time low	$t_{W(CL)}$	20	—	ns	C_L = 50 pF
MDIO setup time to MDC on write	$t_{SU(W)}$	15	—	ns	C_L = 50 pF
MDIO setup time from MDC on write	$t_{H(W)}$	15	—	ns	C_L = 50 pF
MDIO setup time to MDC on read	$t_{SU(R)}$	30	—	ns	C_L = 50 pF on MDC
MDIO hold time from MDC on read	$t_{H(R)}$	0	—	ns	C_L = 50 pF

1. 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.
2. Calculated as $t_C = 1/f$.

6.2.7 SI BOOT TIMING MASTER MODE SPECIFICATIONS

The following table lists the specifications for the signal applied to the SI boot timing master mode input at the SI boot timing master mode pin.

TABLE 6-24: SI BOOT TIMING MASTER MODE CHARACTERISTICS

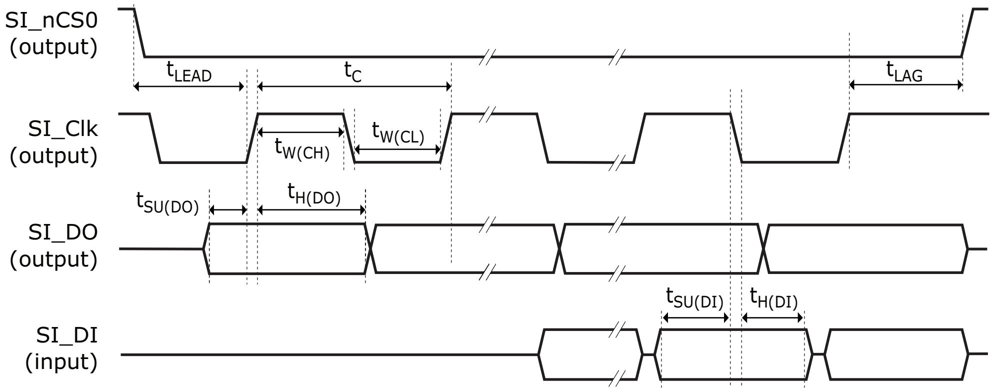
Parameter	Symbol	Minimum	Maximum	Unit	Condition
Clock frequency	f	—	31.15[1]	MHz	—
Clock cycle time	t_C	32	—	ns	—
Clock time high	$t_{W(CH)}$	12	—	ns	—
Clock time low	$t_{W(CL)}$	12	—	ns	—
Clock rise time and fall time	t_R, t_F	—	5	ns	Between $V_{IL(MAX)}$ and $V_{IH(MIN)}$. C_L= 30 pF
SI_DO setup time to clock	$t_{SU(DO)}$	10	—	ns	—
SI_DO hold time from clock	$t_{H(DO)}$	10	—	ns	—
Enable active before first clock	t_{LEAD}	10	—	ns	—
Enable inactive after clock	t_{LAG}	5	—	ns	—
SI_DI setup time to clock	$t_{SU(DIB)}$	20	—	ns	—
SI_DI hold time from clock	$t_{H(DIB)}$	0	—	ns	—

1. Frequency is programmable. The startup frequency is 8.1 MHz.

6.2.8 SI TIMING MASTER MODE SPECIFICATIONS

All serial interface (SI) timing requirements for master mode are specified relative to the input low and input high threshold levels. The following illustration shows the timing parameters and measurement points.

FIGURE 6-3: SI TIMING DIAGRAM FOR MASTER MODE



The following table lists the specifications for the signal applied to the SI timing master mode input at the SI timing master mode pin.

TABLE 6-25: SI TIMING MASTER MODE CHARACTERISTICS

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Clock frequency	f	—	31.25[1]	MHz	—
Clock cycle time	t_C	32	—	ns	—
Clock time high	$t_{W(CH)}$	12	—	ns	—
Clock time low	$t_{W(CL)}$	12	—	ns	—
Clock rise time and fall time	t_R, t_F	—	5	ns	Between $V_{IL(MAX)}$ and $V_{IH(MIN)}$. C_L= 30 pF
SI_DO setup time to clock	$t_{SU(DO)}$	9	—	ns	—
SI_DO hold time from clock	$t_{H(DO)}$	9	—	ns	—
Enable active before first clock	t_{LEAD}	10	—	ns	—
Enable inactive after clock	t_{LAG}	15	—	ns	—
SI_DI sampling time delay[2]	t_{RSD}	0	t_C	ns	—
SI_DI setup time to clock	$t_{SU(DI)}$	18	—	ns	—
SI_DI hold time from clock	$t_{H(DI)}$	0	—	ns	—

[1] Frequency is programmable. The startup frequency is 8.1 MHz.

[2] Delay is programmable in 4 ns steps using register bitfield SIMC:RX_SAMPLE_DLY:RSD.

6.2.9 SI TIMING SLAVE MODE SPECIFICATIONS

All serial interface (SI) slave mode timing requirements are specified relative to the input low and input high threshold levels. The following illustrations show the timing parameters and measurement points for SI input and output data.

FIGURE 6-4: SI INPUT DATA TIMING DIAGRAM FOR SLAVE MODE

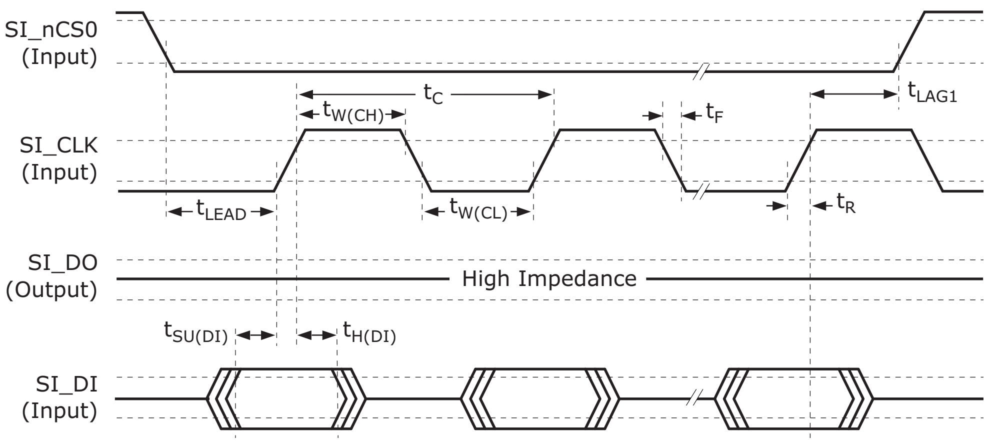
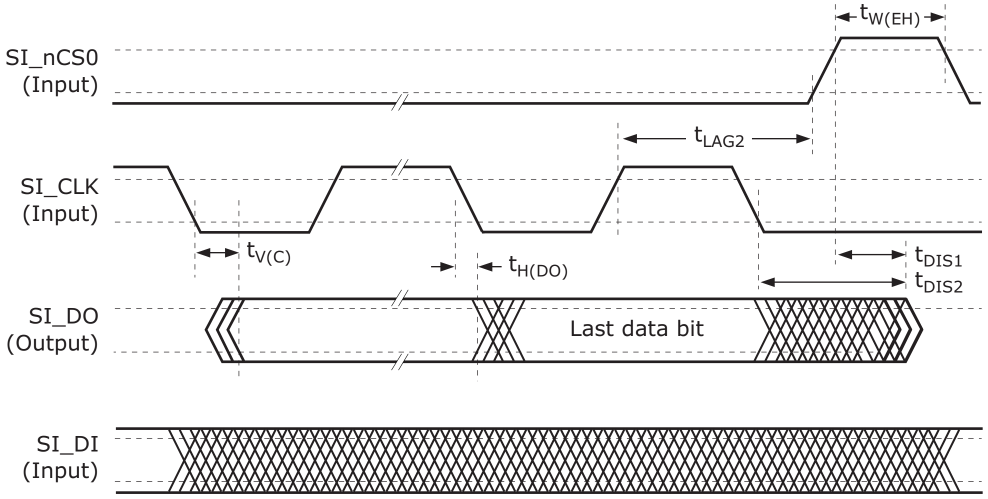


FIGURE 6-5: SI OUTPUT DATA TIMING DIAGRAM FOR SLAVE MODE



The following table lists the specifications for the signal applied to the SI timing slave mode input at the SI timing slave mode pin.

TABLE 6-26: SI TIMING SLAVE MODE CHARACTERISTICS

Parameter	Symbol	Minimum	Maximum	Unit	Condition
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 setup time to clock	$t_{SU(DI)}$	4	—	ns	—
SI_DI hold time from clock	$t_{H(DI)}$	4	—	ns	—
Enable active before first clock	t_{LEAD}	10	—	ns	—
Enable inactive after clock (input cycle)[1]	t_{LAG1}	25	—	ns	—
Enable inactive after clock (output cycle)	t_{LAG2}	See note[2]	—	ns	—
Enable inactive width	$t_{W(EH)}$	20	—	ns	—
SI_DO valid after clock	$t_{V(C)}$	—	25	ns	C_L = 30 pF
SI_DO hold time from clock	$t_{H(DO)}$	0	—	ns	C_L = 0 pF

TABLE 6-26: SI TIMING SLAVE MODE CHARACTERISTICS (CONTINUED)

Parameter	Symbol	Minimum	Maximum	Unit	Condition
SI_DO disable time[3]	T_{DIS1}	—	20	ns	See Figure 161, page 610
SI_DO disable time[3]	T_{DIS2}	—	20	ns	See Figure 161, page 610

1. t_{LAG1} is defined only for write operations to the devices, not for read operations.
2. 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.
3. Pin begins to float when a 300 mV change from the loaded V_{OH} or V_{OL} level occurs.

6.2.10 DDR3/DDR3L SDRAM INPUT SIGNAL SPECIFICATIONS

The following table lists the DDR3 / DDR3L SDRAM input signal specifications.

TABLE 6-27: DDR3 / DDR3L SDRAM INPUT SIGNAL CHARACTERISTICS

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Condition
DDR3 input voltage high	$V_{IH(AC)}$	DDR_V_{REF} + 0.175	—	V_{DD_IODDR} + 0.3	V	—
DDR3 input voltage low	$V_{IL(AC)}$	–0.3	—	DDR_V_{REF} – 0.175	V	—
DDR3 differential input voltage	$V_{ID(AC)}$	0.4	—	V_{DD_IODDR}	V	—
DDR3 differential crospoint voltage	$V_{IX(AC)}$	0.5 × V_{DD_IODDR} – 0.175	—	0.5 × V_{DD_IODDR} + 0.175	V	—
DDR3L input voltage high	$V_{IH(AC)}$	DDR_V_{REF} + 0.160	—	V_{DD_IODDR} + 0.3	V	—
DDR3L input voltage low	$V_{IL(AC)}$	–0.3	—	DDR_V_{REF} – 0.160	V	—
DDR3L differential input voltage	$V_{ID(AC)}$	0.4	—	V_{DD_IODDR}	V	—
DDR3L differential crospoint voltage	$V_{IX(AC)}$	0.5 × V_{DD_IODDR} – 0.160	—	0.5 × V_{DD_IODDR} + 0.160	V	—
DDR_DQ[7:0] input setup time relative to DDR_LDQS[1]	t_{DS}	350	—	—	ps	—
DDR_DQ[15:8] input setup time relative to DDR_UDQS[1]	t_{DS}	350	—	—	ps	—
DDR_DQ[7:0] input hold time relative to DDR_LDQS[1]	t_{DH}	250	—	—	ps	—
DDR_DQ[15:8] input hold time relative to DDR_UDQS[1]	t_{DH}	250	—	—	ps	—
Quarter period offset from clock edge	t_{90}	—	0.25 x t_{CLK}	—	ns	t_{CLK} = 3.2 ns

1. These requirements are dependent on the operation frequency of the DDR SDRAM interface. Stated limits are for DDR3-800,

6.2.11 DDR3/DDR3L SDRAM OUTPUT SIGNAL SPECIFICATIONS

The following table lists the DDR3 / DDR3L SDRAM output signal specifications.

TABLE 6-28: DDR3 / DDR3L SDRAM OUTPUT SIGNAL CHARACTERISTICS

Parameter	Symbol	Minimum	Typical	Maximum	Unit
DDR_CK cycle time 312.5 MHz (DDR800)[1]	t_{CLK}	—	3.20	—	ns
DDR_CK/CKn duty cycle		48	—	52	%
DDR_A, DDR_BA, DDR_CKE, DDR_nRAS, DDR_nCAS, and DDR_nWE output setup time relative to DDR_CK/CKn	t_{AS}	700	—	—	ps
DDR_A, DDR_BA, DDR_CKE, DDR_nRAS, DDR_nCAS, and DDR_nWE output hold time relative to DDR_CK/CKn	t_{AH}	800	—	—	ps
DDR_CK/CKn to DDR_DQS skew	t_{CLKDQS}	–400	—	400	ps
DDR_DQ[7:0]/DDR_LDM output setup time relative to DDR_LDQS	t_{DS}	400	—	—	ps
DDR_DQ[15:8]/DDR_UDM output setup time relative to DDR_UDQS	t_{DS}	400	—	—	ps
DDR_DQ[7:0]/DDR_LDM output hold time relative to DDR_LDQS	t_{DH}	500	—	—	ps
DDR_DQ[15:8]/DDR_UDM output hold time relative to DDR_UDQS	t_{DH}	500	—	—	ps
DDR_xDQS preamble start	t_{DQSPRE}	0.4 x t_{CLK}	—	0.6 x t_{CLK}	ps
DDR_xDQS postamble end	t_{DQSPST}	0.4 x t_{CLK}	—	0.6 x t_{CLK}	ps

[1] Timing reference is DDR_CK/DDR_CKn crossing ±0.1 V.

6.2.12 JTAG INTERFACE SPECIFICATIONS

The following table lists the JTAG interface specifications.

TABLE 6-29: JTAG INTERFACE CHARACTERISTICS

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Condition
JTAG_TCK frequency	f	—	—	10	MHz	—
JTAG_TCK cycle time	t_C	100	—	—	ns	—
JTAG_TCK high time	$t_{W(CH)}$	40	—	—	ns	—
JTAG_TCK low time	$t_{W(CL)}$	40	—	—	ns	—
Setup time to JTAG_TCK rising	t_{SU}	10	—	—	ns	—
Hold time from JTAG_TCK rising	t_H	10	—	—	ns	—
JTAG_TDO valid after JTAG_TCK falling	$t_{V(c)}$		—	28	ns	C_L = 10 pF
JTAG_TDO hold time from JTAG_TCK falling	$t_{H(TDO)}$	0	—		ns	C_L = 0 pF
JTAG_TDO disable time[1]	t_{DIS}		—	30	ns	See Figure 171, page 651
JTAG_nTRST time low	$t_{W(TL)}$	30	—		ns	

[1] The pin begins to float when a 300 mV change from the actual V_{OH}/V_{OL} level occurs.

6.2.13 SERIAL I/O TIMING SPECIFICATIONS

The following table lists the serial I/O timing specifications.

TABLE 6-30: SERIAL I/O TIMING CHARACTERISTICS

Parameter	Symbol	Minimum	Typical	Maximum	Unit
Clock frequency[1]	f	—	—	25	MHz
SG0_CLK clock pulse width	$t_{W(CLK)}$	40	—	60	%
SG0_DO valid after clock falling	$t_{V(DO)}$	—	—	6	ns
SG0_DO hold time from clock falling	$t_{H(DO)}$	—	—	6	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	$t_{SU(DI)}$	25	—	—	ns
SG0_DI hold time from clock	$t_{H(DI)}$	4	—	—	ns

[1] SIO clock frequency is programmable.

6.2.14 RECOVERED CLOCK OUTPUTS SPECIFICATIONS

The following table lists the recovered clock outputs specifications.

TABLE 6-31: RECOVERED CLOCK OUTPUTS CHARACTERISTICS

Parameter	Symbol	Minimum	Maximum	Unit	Condition
RECO_CLK[1:0] clock frequency	f	—	156.25	MHz	—
Clock duty cycle	t_C	40	60	%	Measured at 50% threshold
RECO_CLK[1:0] rise time and fall time	t_R t_F	—	1.5	ns	
Squelching delay from SGMII signal to RECO_CLK[1:0]	—	—	200	ns	Squelch enabled
RECO_CLK[1:0] peak-to-peak jitter, bandwidth between 12 kHz and 10 MHz[1], 60 second gate time	—	—	200	ps	Jitter-free input to SerDes Rx
RECO_CLK[1:0] peak-to-peak jitter, bandwidth between 10 MHz and 80 MHz[1], 60 second gate time	—	—	200	ps	Jitter-free input to SerDes Rx

[1] Maximum jitter on the recovered signal.

6.2.15 TWO-WIRE SERIAL INTERFACE SPECIFICATIONS

This section contains information about the device two-wire serial slave interface. The following table lists the two-wire serial timing specifications.

TABLE 6-32: TWO-WIRE SERIAL TIMING CHARACTERISTICS

Parameter	Symbol	Standard	Standard	Fast Mode	Fast Mode	Unit	
		Minimum	Maximum	Minimum	Maximum		Condition
TWI_SCL clock frequency	f	—	100	—	400	kHz	—
TWI_SCL low period	t_{LOW}	4.7	—	1.3	—	µs	—
TWI_SCL high period	t_{HIGH}	4.0	—	0.6	—	µs	—
TWI_SCL and TWI_SDA rise time		—	1000	—	300	ns	—
TWI_SCL and TWI_SDA fall time		—	300	—	300	ns	—

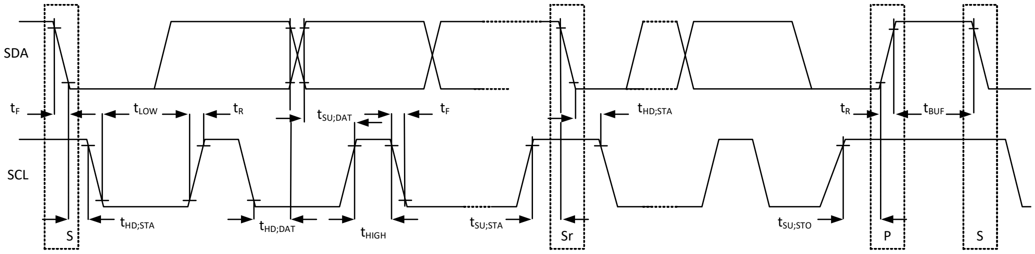
TABLE 6-32: TWO-WIRE SERIAL TIMING CHARACTERISTICS (CONTINUED)

Parameter	Symbol	Standard	Standard	Fast Mode	Fast Mode	Unit	
		Minimum	Maximum	Minimum	Maximum		Condition
TWI_SDA setup time to TWI_SCL fall	t_{SU_DAT}	250		100	300	ns	—
TWI_SDA hold time to TWI_SCL fall[1]	t_{HD_DAT}	300	3450	300	900	ns	300 ns delay enabled in ICPU_CFG::TWI_-CONFIG.register
Setup time for repeated START condition	t_{SU_STA}	4.7	—	0.6	—	µs	—
Hold time after repeated START condition	t_{HD_STA}	4.0	—	0.6	—	µs	—
Bus free time between STOP and START conditions	t_{BUF}	4.7	—	1.3	—	µs	—
Clock to valid data out[2]	t_{VD_DAT}	300	—	300	—	ns	—
Pulse width of spike suppressed by input filter on TWI_SCL or TWI_SDA[3]	—	4	128	4	128	ns	—

1. 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.
2. 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 given value represents the recommended minimum value, which is enabled in ICPU_CFG::TWI_CONFIG.TWI_DELAY_ENABLE.
3. Configurable in the register ICPU_CFG::TWI_SPIKE_FILTER_CFG. The default register value results in spike suppression up to 4 ns. The recommended register value is 12, resulting in spike suppression up to 52 ns.

The following illustration shows the two-wire serial interface timing.

FIGURE 6-6: TWO-WIRE SERIAL INTERFACE TIMING



S = START, P = STOP, and Sr = repeated START.

6.2.16 IEEE1588 TIME TICK OUTPUT SPECIFICATIONS

The following table lists the IEEE1588 Time Tick Output specifications.

TABLE 6-33: IEEE1588 TIME TICK OUTPUT CHARACTERISTICS

Parameter	Symbol	Minimum	Maximum	Unit	Condition
PTP[3:0] frequency[1]	f	—	25	MHz	—
Clock duty cycle[2]	—	45	55	%	Measured at 50% threshold
PTP[3:0] rise time and fall time	t_R, t_F	1	—	ns	20% to 80% threshold

TABLE 6-33: IEEE1588 TIME TICK OUTPUT CHARACTERISTICS

Parameter	Symbol	Minimum	Maximum	Unit	Condition
PTP[3:0] peak-to-peak jitter, bandwidth between 12 kHz and 10 MHz[3]	—	—	200	ps	—
PTP[3:0] peak-to-peak jitter, bandwidth between 10 MHz and 80 MHz[3]	—	—	200	ps	—

[1] Frequency is programmable.

[2] Both high and low periods are configured to the identical value using DEVCPU_PTP::PIN_WF_HIGH_PERIOD and DEVCPU_PTP::PIN_WF_LOW_PERIOD.

[3] Timing corrections performed by register writes and high/low periods that do not match a fixed number of 156.25 MHz clock cycles will create jitter up to 3.5 ns.

6.3 Current and Power Consumption

The following tables lists the current and power consumption for the VSC7513 device respectively.

TABLE 6-34: CURRENT AND POWER CONSUMPTION

Parameter	Symbol	Typical	Maximum (TJ_MAX = 110 °C)	Maximum (TJ_MAX = 125 °C)	Unit
V_{DD} operating current, 1.0 V	I_{DD}	0.9	1.5	1.8	A
V_{DD_A} operating current, 1.0 V	I_{DD_A}	0.26	0.26	0.26	A
V_{DD_VS} operating current, 1.0 V	I_{DD_VS}	0.15	0.15	0.15	A
V_{DD_AL} operating current, 1.0 V	I_{DD_AL}	0.08	0.1	0.1	A
V_{DD_AH} operating current, 2.5 V	I_{DD_AH}	0.45	0.58	0.58	A
V_{DD_IODDR} operating current[1], 1.35 V or 1.5 V	I_{DD_IODDR}	0.1	0.1	0.1	A
V_{DD_IO} operating current[2], 2.5 V	I_{DD_IO}	0.02	0.02	0.02	A
Total Power	—	2.7	3.8	4.2	W
Saving per disabled CuPHY port	—	0.4	0.55	0.55	W
Saving per disabled SerDes1G port	—	0.02	0.03	0.03	W
Saving per disabled SerDes6G port	—	0.07	0.09	0.09	W

[1] DDR3 on-die termination is disabled.

[2] Unloaded pins.

6.4 Operating Conditions

The following table lists the recommended operating conditions for the VSC7513 device.

TABLE 6-35: RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Minimum	Typical	Maximum	Unit
Power supply voltage for core supply	V_{DD}	0.95	1.00	1.05	V
Power supply voltage for analog circuits	V_{DD_A}	0.95	1.00	1.05	V
Power supply voltage for 1G and 6G interfaces, 1.0V	V_{DD_VS}	0.95	1.00	1.05	V
Power supply voltage for 1G and 6G interfaces, 1.2V	V_{DD_VS}	1.14	1.2	1.26	V
Power supply voltage for copper PHY analog circuits	V_{DD_AL}	0.95	1.0	1.05	V
Power supply voltage for copper PHY interface	V_{DD_AH}	2.375	2.5	2.625	V
Power supply voltage for DDR3 interface	V_{DD_IODDR}	1.425	1.5	1.575	V

TABLE 6-35: RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Minimum	Typical	Maximum	Unit
Power supply voltage for DDR3L interface	V_{DD_IODDR}	1.28	1.35	1.45	V
Power supply voltage for GPIO and miscellaneous I/O	V_{DD_IO}	2.375	2.5	2.625	V
Operating temperature[1]	T	–40		125	°C

[1] Minimum specification is ambient temperature, and the maximum is junction temperature.

6.4.1 POWER SUPPLY SEQUENCING

During power on and off, V_{DD_A}, and V_{DD_VS} must never be more than 300 mV above VDD.

V_{DD_VS} must be powered, even if the associated interfaces are not used. These power supplies must not remain at ground or be left floating.

A maximum delay of 100 ms from V_{DD_IODDR} to V_{DD} is recommended. There is no requirement from V_{DD} to V_{DD_IODDR}.

The V_{DD_IODDR} supply can remain at ground or be left floating if not used. If V_{DD_IODDR} is grounded, DDR_Vref must also be grounded.

There are no sequencing requirements for V_{DD_IO}.

The nRESET and JTAG_nTRST inputs must be held low until all power supply voltages have reached their recommended operating condition values.

6.5 Stress Ratings

This section contains the stress ratings for the VSC7513 device.

Warning Stresses listed in the following table may be applied to devices one at a time without causing permanent damage. Functionality at or exceeding the values listed is not implied. Exposure to these values for extended periods may affect device reliability.

TABLE 6-36: STRESS RATINGS

Parameter	Symbol	Minimum	Maximum	Unit	Condition
Power supply voltage for core supply	V_{DD}	–0.3	1.10	V	Design
Power supply voltage for analog circuits	V_{DD_A}	–0.3	1.10	V	Design
Power supply voltage for 1G and 6G interfaces	V_{DD_VS}	–0.3	1.32	V	Design
Power supply voltage for copper PHY analog circuits	V_{DD_AL}	–0.3	1.10	V	Design
Power supply voltage for copper PHY interface	V_{DD_AH}	–0.3	2.75	V	Design
Power supply voltage for DDR I/O buffers	V_{DD_IODDR}	–0.3	1.98	V	Design
Power supply voltage for GPIO and miscellaneous I/O	V_{DD_IO}	–0.3	2.75	V	Design
Storage temperature	T_S	–55	125	°C	Design
Electrostatic discharge voltage, charged device model	V_{ESD_CDM}	–250	250	V	
Electrostatic discharge voltage, human body model	V_{ESD_HBM}	See note[1]		V	

[1] This device has completed all required testing as specified in the JEDEC standard JESD22-A114, *Electrostatic Discharge (ESD) Sensitivity Testing Human Body Model (HBM)*, and complies with a Class 2 rating. The definition of Class 2 is any part that passes an ESD pulse of 2000V, but fails an ESD pulse of 4000V.

Note: This device can be damaged by electrostatic discharge (ESD) voltage. Microchip recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures may adversely affect reliability of the device.

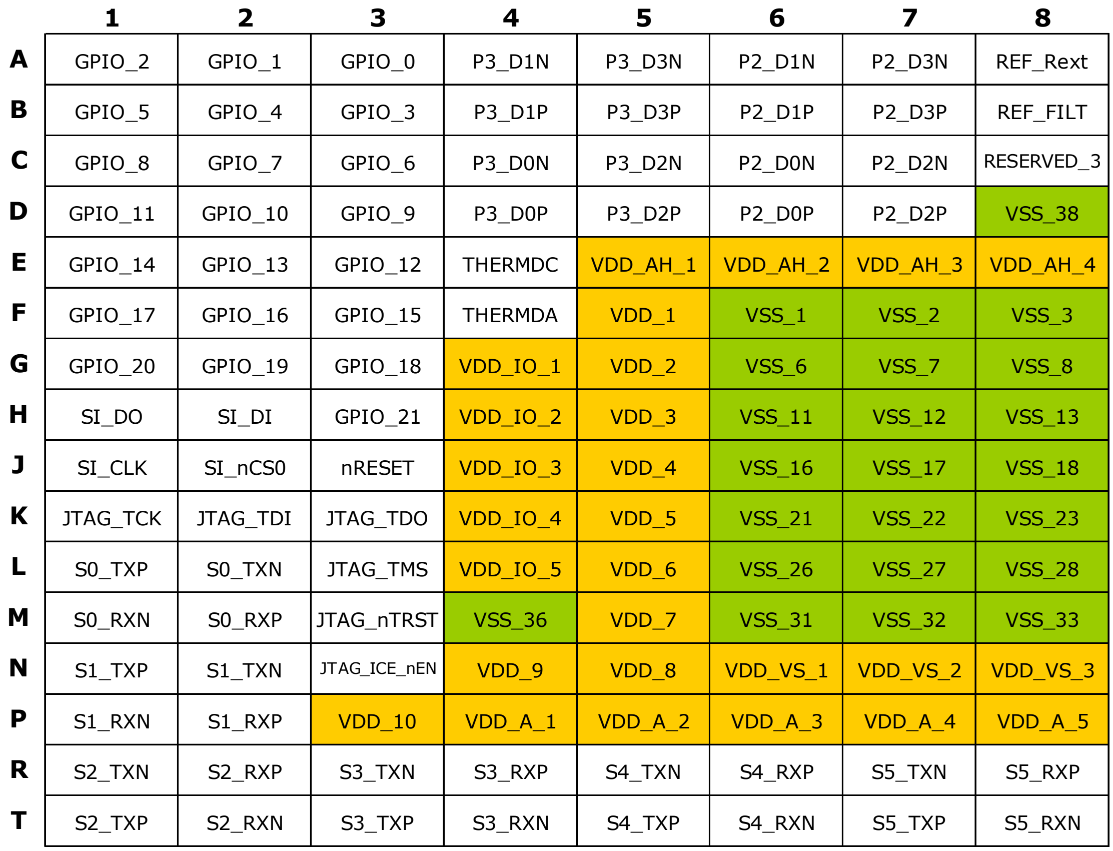
7.0 PIN DESCRIPTIONS

The VSC7513 device has 256 pins, which are described in this section. The pin information is also provided in the attached file, which you can copy, sort, or manipulate, as desired.

7.1 Pin Diagram

The following illustration is a representation of the VSC7513 device, as seen from the bottom view.

FIGURE 7-1: PIN DIAGRAM, TOP LEFT



	1	2	3	4	5	6	7	8
A	GPIO_2	GPIO_1	GPIO_0	P3_D1N	P3_D3N	P2_D1N	P2_D3N	REF_Rext
B	GPIO_5	GPIO_4	GPIO_3	P3_D1P	P3_D3P	P2_D1P	P2_D3P	REF_FILT
C	GPIO_8	GPIO_7	GPIO_6	P3_D0N	P3_D2N	P2_D0N	P2_D2N	RESERVED_3
D	GPIO_11	GPIO_10	GPIO_9	P3_D0P	P3_D2P	P2_D0P	P2_D2P	VSS_38
E	GPIO_14	GPIO_13	GPIO_12	THERMDC	VDD_AH_1	VDD_AH_2	VDD_AH_3	VDD_AH_4
F	GPIO_17	GPIO_16	GPIO_15	THERMDA	VDD_1	VSS_1	VSS_2	VSS_3
G	GPIO_20	GPIO_19	GPIO_18	VDD_IO_1	VDD_2	VSS_6	VSS_7	VSS_8
H	SI_DO	SI_DI	GPIO_21	VDD_IO_2	VDD_3	VSS_11	VSS_12	VSS_13
J	SI_CLK	SI_nCS0	nRESET	VDD_IO_3	VDD_4	VSS_16	VSS_17	VSS_18
K	JTAG_TCK	JTAG_TDI	JTAG_TDO	VDD_IO_4	VDD_5	VSS_21	VSS_22	VSS_23
L	S0_TXP	S0_TXN	JTAG_TMS	VDD_IO_5	VDD_6	VSS_26	VSS_27	VSS_28
M	S0_RXN	S0_RXP	JTAG_nTRST	VSS_36	VDD_7	VSS_31	VSS_32	VSS_33
N	S1_TXP	S1_TXN	JTAG_ICE_nEN	VDD_9	VDD_8	VDD_VS_1	VDD_VS_2	VDD_VS_3
P	S1_RXN	S1_RXP	VDD_10	VDD_A_1	VDD_A_2	VDD_A_3	VDD_A_4	VDD_A_5
R	S2_TXN	S2_RXP	S3_TXN	S3_RXP	S4_TXN	S4_RXP	S5_TXN	S5_RXP
T	S2_TXP	S2_RXN	S3_TXP	S3_RXN	S4_TXP	S4_RXN	S5_TXP	S5_RXN

FIGURE 7-2: PIN DIAGRAM, TOP RIGHT

9	10	11	12	13	14	15	16	
P1_D1N	P1_D3N	P0_D1N	P0_D3N	DDR_Vref	DDR_DQ12	DDR_DQ14	DDR_DQ15	A
P1_D1P	P1_D3P	P0_D1P	P0_D3P	DDR_UDQSn	DDR_UDQS	DDR_DQ13	DDR_DQ9	B
P1_D0N	P1_D2N	P0_D0N	P0_D2N	DDR_DQ10	DDR_DQ8	DDR_DQ11	DDR_UDM	C
P1_D0P	P1_D2P	P0_D0P	P0_D2P	DDR_A5	DDR_A7	RESERVED_2	DDR_ODT	D
VDD_AH_5	VDD_AL_1	VDD_AL_2	RESERVED_1	DDR_A13	DDR_A9	DDR_nCS1	DDR_BA0	E
VSS_4	VSS_5	VDD_20	RESERVED_0	DDR_A2	DDR_A3	DDR_A0	DDR_BA2	F
VSS_9	VSS_10	VDD_19	VDD_IODDR_1	DDR_nCS0	DDR_nCAS	DDR_nWE	DDR_CK	G
VSS_14	VSS_15	VDD_18	VDD_IODDR_2	DDR_nRAS	DDR_A10	DDR_CKn	DDR_A1	H
VSS_19	VSS_20	VDD_17	VDD_IODDR_3	DDR_A12	DDR_A15	DDR_A11	DDR_A14	J
VSS_24	VSS_25	VDD_16	VDD_IODDR_4	DDR_A6	DDR_BA1	DDR_A8	DDR_A4	K
VSS_29	VSS_30	VDD_15	VDD_IODDR_5	DDR_DQ6	DDR_DQ4	DDR_CKE	DDR_DQ5	L
VSS_34	VSS_35	VDD_14	VSS_37	DDR_LDQS	DDR_LDQSn	DDR_DQ7	DDR_DQ1	M
VDD_VS_4	VDD_VS_5	VDD_13	VDD_12	DDR_DQ0	DDR_DQ2	DDR_DQ3	DDR_LDM	N
VDD_A_6	VDD_A_7	VDD_A_8	VDD_A_9	VDD_11	SERDES_Rext_0	SERDES_Rext_1	DDR_Rext	P
S6_TXN	S6_RXP	RESERVED_39	RESERVED_42	S8_TXN	S8_RXP	CLKOUTPLL_N	REFCLK_P	R
S6_TXP	S6_RXN	RESERVED_40	RESERVED_41	S8_TXP	S8_RXN	CLKOUTPLL_P	REFCLK_N	T

7.2 Pins by Function

This section describes the functional pin descriptions for the VSC7513 device.

The following table lists the definitions for the pin type symbols.

TABLE 7-1: PIN TYPE SYMBOL DEFINITIONS

Symbol	Pin Type	Description
A	Analog input	Analog input for sensing variable voltage levels
ABIAS	Analog bias	Analog bias pin
DIFF	Differential	Differential signal pair
I	Input	Input signal
O	Output	Output signal
I/O	Bidirectional	Bidirectional input or output signal
OZ	3-state output	Output
LVDS	Input or output	Low voltage differential signal
LVCMOS	Input or output	Low voltage CMOS signal
3V		3.3V-tolerant
ST	Schmitt-trigger	Input has Schmitt-trigger circuitry
TD	Termination differential	Internal differential termination

The following table lists the functional pin descriptions for the VSC7513 device.

TABLE 7-2: PINS BY FUNCTION

Group	Name	Numbe r	I/O	Type	Level	Description
ANALOG	DDR_Rext	P16	A	ABIAS	Analog	External DDR impedance calibration. Connect the pin through an external 240 Ω ±1% resistor to ground.
ANALOG	DDR_Vref	A13	A	ABIAS	Analog	Input reference voltage. Provides the input switching reference voltage to the SSTL DDR signals.
ANALOG	REF_FILT	B8	A	ABIAS	Analog	Copper media reference filter pin. Connect a 1.0 μF external capacitor between pin and ground.
ANALOG	REF_Rext	A8	A	ABIAS	Analog	Copper media reference external pin. Connect a 2.0 kΩ (1%) resistor between pin and ground.
ANALOG	SERDES_Rext_0	P14	A	Analog	Analog	Analog bias calibration. Connect an external 620 Ω ±1% resistor between SERDES_Rext_1 and SERDES_Rext_0.
ANALOG	SERDES_Rext_1	P15	A		Analog	Analog bias calibration. Connect an external 620 Ω ±1% resistor between SERDES_Rext_1 and SERDES_Rext_0.
ANALOG	THERMDA	F4	A	Analog	Analog	Thermal diode anode (p-junction).
ANALOG	THERMDC	E4	A	Analog	Analog	Thermal diode cathode (n-junction). Connected on-die to VSS.
CLOCK	CLKOUTPLL_N	R15	O		LVDS	PLL clock out. Leave floating if not used.
CLOCK	CLKOUTPLL_P	T15	O		LVDS	PLL clock out. Leave floating if not used.
CLOCK	REFCLK_N	T16	I	TD	LVDS	Reference clock inputs. The inputs can be either differential or single-ended. In differential mode (LVDS), REFCLK_P is the true part of the differential signal, and REFCLK_N/REFCLK_N is the complement part of the differential signal. In single-ended mode (LVCMOS), REFCLK_P is used as single-ended LVTTL input. REFCLK_N should be left floating, and the PLL registers must be configured for singleended operation. Required applied frequency depends on REFCLK_CONF[2:0] input state.

CLOCK	REFCLK_P	R16	I	TD	LVDS	Reference clock inputs. The inputs can be either differential or single-ended. In differential mode (LVDS), REFCLK_P/ is the true part of the differential signal, and REFCLK_N/REFCLK_N is the complement part of the differential signal. In single-ended mode (LVCMOS), REFCLK_P is used as single-ended LVTTL input. REFCLK_N should be left floating, and the PLL registers must be configured for singleended operation. Required applied frequency depends on REFCLK_CONF[2:0] input state.
DDR	DDR_A0	F15	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_A1	H16	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_A10	H14	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_A11	J15	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_A12	J13	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_A13	E13	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_A14	J16	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_A15	J14	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_A2	F13	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_A3	F14	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_A4	K16	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.

DDR	DDR_A5	D13	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_A6	K13	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_A7	D14	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_A8	K15	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_A9	E14	O		SSTL	SDRAM address outputs. Provide row and column addresses to the SDRAM.
DDR	DDR_BA0	E16	O		SSTL	SDRAM bank address outputs. DDR_BA[2:0] define to which bank an ACTIVE, READ, WRITE, or PRECHARGE command is applied.
DDR	DDR_BA1	K14	O		SSTL	SDRAM bank address outputs. DDR_BA[2:0] define to which bank an ACTIVE, READ, WRITE, or PRECHARGE command is applied.
DDR	DDR_BA2	F16	O		SSTL	SDRAM bank address outputs. DDR_BA[2:0] define to which bank an ACTIVE, READ, WRITE, or PRECHARGE command is applied.
DDR	DDR_CK	G16	O	DIFF	SSTL	SDRAM differential clock. Differential clock to external SDRAM. DDR_CK is the true part of the differential signal. DDR_CKn is the complement part.
DDR	DDR_CKE	L15	O		SSTL	SDRAM clock enable. 0: Disables clock in external SDRAM. 1: Enables clock in external SDRAM.
DDR	DDR_CKn	H15	O	DIFF	SSTL	SDRAM differential clock. Differential clock to external SDRAM. DDR_CK is the true part of the differential signal. DDR_CKn is the complement part.
DDR	DDR_DQ0	N13	I/O		SSTL	SDRAM data bus.
DDR	DDR_DQ1	M16	I/O		SSTL	SDRAM data bus.
DDR	DDR_DQ10	C13	I/O		SSTL	SDRAM data bus.
DDR	DDR_DQ11	C15	I/O		SSTL	SDRAM data bus.
DDR	DDR_DQ12	A14	I/O		SSTL	SDRAM data bus.
DDR	DDR_DQ13	B15	I/O		SSTL	SDRAM data bus.
DDR	DDR_DQ14	A15	I/O		SSTL	SDRAM data bus.
DDR	DDR_DQ15	A16	I/O		SSTL	SDRAM data bus.

DDR	DDR_DQ2	N14	I/O		SSTL	SDRAM data bus.
DDR	DDR_DQ3	N15	I/O		SSTL	SDRAM data bus.
DDR	DDR_DQ4	L14	I/O		SSTL	SDRAM data bus.
DDR	DDR_DQ5	L16	I/O		SSTL	SDRAM data bus.
DDR	DDR_DQ6	L13	I/O		SSTL	SDRAM data bus.
DDR	DDR_DQ7	M15	I/O		SSTL	SDRAM data bus.
DDR	DDR_DQ8	C14	I/O		SSTL	SDRAM data bus.
DDR	DDR_DQ9	B16	I/O		SSTL	SDRAM data bus.
DDR	DDR_LDM	N16	O		SSTL	SDRAM data mask outputs. DDR_LDM and DDR_UDM are mask signals for data written to the SDRAM. DDR_LDM corresponds to data on DDR_DQ[7:0], and DDR_UDM corresponds to data on DDR_DQ[15:8].
DDR	DDR_LDQS	M13	I/O	DIFF, TD	SSTL	SDRAM differential data strobes. Bidirectional differential signal that follows data direction. Used by SDRAM to capture data on write operations. Edge-aligned with data from SDRAM during read operations and used by the device to capture data. DDR_LDQS corresponds to data on DDR_DQ[7:0], and DDR_UDQS corresponds to data on DDR_DQ[15:8].
DDR	DDR_LDQSn	M14	I/O	DIFF, TD	SSTL	SDRAM differential data strobes. Bidirectional differential signal that follows data direction. Used by SDRAM to capture data on write operations. Edge-aligned with data from SDRAM during read operations and used by the device to capture data. DDR_LDQS corresponds to data on DDR_DQ[7:0], and DDR_UDQS corresponds to data on DDR_DQ[15:8].
DDR	DDR_nCAS	G14	O		SSTL	SDRAM command outputs. DDR_nRAS, DDR_nCAS, and DDR_nWE (along with DDR_nCS) define the command being entered.
DDR	DDR_nCS0	G13	O		SSTL	
DDR	DDR_nCS1	E15	O		SSTL	
DDR	DDR_nRAS	H13	O		SSTL	SDRAM command outputs. DDR_nRAS, DDR_nCAS, and DDR_nWE (along with DDR_nCS) define the command being entered.

DDR	DDR_nWE	G15	O		SSTL	SDRAM command outputs. DDR_nRAS, DDR_nCAS, and DDR_nWE (along with DDR_nCS) define the command being entered.
DDR	DDR_ODT	D16	O		SSTL	Control signal for the attached DDR3/ DDR3L SDRAM devices ondie termination.
DDR	DDR_UDM	C16	B		SSTL	SDRAM data mask outputs. DDR_LDM and DDR_UDM are mask signals for data written to the SDRAM. DDR_LDM corresponds to data on DDR_DQ[7:0], and DDR_UDM corresponds to data on DDR_DQ[15:8].
DDR	DDR_UDQS	B14	I/O	DIFF, TD	SSTL	SDRAM differential data strobes. Bidirectional differential signal that follows data direction. Used by SDRAM to capture data on write operations. Edge-aligned with data from SDRAM during read operations and used by the device to capture data. DDR_LDQS corresponds to data on DDR_DQ[7:0], and DDR_UDQS corresponds to data on DDR_DQ[15:8].
DDR	DDR_UDQSn	B13	I/O	DIFF, TD	SSTL	SDRAM differential data strobes. Bidirectional differential signal that follows data direction. Used by SDRAM to capture data on write operations. Edge-aligned with data from SDRAM during read operations and used by the device to capture data. DDR_LDQS corresponds to data on DDR_DQ[7:0], and DDR_UDQS corresponds to data on DDR_DQ[15:8].
DDR	RESERVED_2	D15	O		SSTL	Leave floating
GPIO	GPIO_0	A3	I/O	PU,ST,3V	CMOS_2.5	General-purpose input/output/strapping
GPIO	GPIO_1	A2	I/O	PU,ST,3V	CMOS_2.5	General-purpose input/output/strapping
GPIO	GPIO_2	A1	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.
GPIO	GPIO_3	B3	I/O	PU,ST,3V	CMOS_2.5	General-purpose input/output/strapping.
GPIO	GPIO_4	B2	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.
GPIO	GPIO_5	B1	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.

GPIO	GPIO_6	C3	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.
GPIO	GPIO_7	C2	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.
GPIO	GPIO_8	C1	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.
GPIO	GPIO_9	D3	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.
GPIO	GPIO_10	D2	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.
GPIO	GPIO_11	D1	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.
GPIO	GPIO_12	E3	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.
GPIO	GPIO_13	E2	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.
GPIO	GPIO_14	E1	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.
GPIO	GPIO_15	F3	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.
GPIO	GPIO_16	F2	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.
GPIO	GPIO_17	F1	I/O	PU,ST,3V	CMOS_2.5	General-purpose inputs and outputs.
GPIO	GPIO_18	G3	I/O	PU,ST,3V	CMOS_2.5	General-purpose input/output/strapping.
GPIO	GPIO_19	G2	I/O	PU,ST,3V	CMOS_2.5	General-purpose input/output/strapping.
GPIO	GPIO_20	G1	I/O	PU,3V	CMOS_2.5	General-purpose input/output/strapping.
GPIO	GPIO_21	H3	I/O	PU,3V	CMOS_2.5	General-purpose input/output/strapping.
JTAG	JTAG_ICE_nEN	N3	I	PU,ST,3V	CMOS_2.5	0: Enables VCore-III debug interface over the JTAG interface. 1: Enables normal JTAG I/O over the JTAG interface. JTAG_ICE_nEN should be pulled either high or low using a resistor of 10 kΩ or less to enable the desired operating mode.
JTAG	JTAG_nTRST	M3	I	PU,ST,3V	CMOS_2.5	JTAG test reset, active low. For normal device operation, JTAG_nTRST should be pulled low.
JTAG	JTAG_TCK	K1	I	PU,ST,3V	CMOS_2.5	JTAG clock.
JTAG	JTAG_TDI	K2	I	PU,ST,3V	CMOS_2.5	JTAG test data in.

JTAG	JTAG_TDO	K3	O	OZ, 3V	CMOS_2.5	JTAG test data out.
JTAG	JTAG_TMS	L3	I	PU,ST,3V	CMOS_2.5	JTAG test mode select.
MISC	nRESET	J3	I	PU,ST,3V	CMOS_2.5	Global device reset, active low.
PHY	P0_D0N	C11	I/O	ADIFF	Analog 2.5	Tx/Rx channel A negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 2.
PHY	P0_D0P	D11	I/O	ADIFF	Analog 2.5	Tx/Rx channel A positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 1.
PHY	P0_D1N	A11	I/O	ADIFF	Analog 2.5	Tx/Rx channel B negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 6.
PHY	P0_D1P	B11	I/O	ADIFF	Analog 2.5	Tx/Rx channel B positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 3.

PHY	P0_D2N	C12	I/O	ADIFF	Analog 2.5	Tx/Rx channel C negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 5 (pins not used in 10/100 Mbps modes).
PHY	P0_D2P	D12	I/O	ADIFF	Analog 2.5	Tx/Rx channel C positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 4 (pins not used in 10/100 Mbps modes).
PHY	P0_D3N	A12	I/O	ADIFF	Analog 2.5	Tx/Rx channel D negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 8 (pins not used in 10/100 Mbps modes).
PHY	P0_D3P	B12	I/O	ADIFF	Analog 2.5	Tx/Rx channel D positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 7 (pins not used in 10/100 Mbps modes).
PHY	P1_D0N	C9	I/O	ADIFF	Analog 2.5	Tx/Rx channel A negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 2.

PHY	P1_D0P	D9	I/O	ADIFF	Analog 2.5	Tx/Rx channel A positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 1.
PHY	P1_D1N	A9	I/O	ADIFF	Analog 2.5	Tx/Rx channel B negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 6.
PHY	P1_D1P	B9	I/O	ADIFF	Analog 2.5	Tx/Rx channel B positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 3.
PHY	P1_D2N	C10	I/O	ADIFF	Analog 2.5	Tx/Rx channel C negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 5 (pins not used in 10/100 Mbps modes).
PHY	P1_D2P	D10	I/O	ADIFF	Analog 2.5	Tx/Rx channel C positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 4 (pins not used in 10/100 Mbps modes).

PHY	P1_D3N	A10	I/O	ADIFF	Analog 2.5	Tx/Rx channel D negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 8 (pins not used in 10/100 Mbps modes).
PHY	P1_D3P	B10	I/O	ADIFF	Analog 2.5	Tx/Rx channel D positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 7 (pins not used in 10/100 Mbps modes).
PHY	P2_D0N	C6	I/O	ADIFF	Analog 2.5	Tx/Rx channel A negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 2.
PHY	P2_D0P	D6	I/O	ADIFF	Analog 2.5	Tx/Rx channel A positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 1.
PHY	P2_D1N	A6	I/O	ADIFF	Analog 2.5	Tx/Rx channel B negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 6.

PHY	P2_D1P	B6	I/O	ADIFF	Analog 2.5	Tx/Rx channel B positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 3.
PHY	P2_D2N	C7	I/O	ADIFF	Analog 2.5	Tx/Rx channel C negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 5 (pins not used in 10/100 Mbps modes).
PHY	P2_D2P	D7	I/O	ADIFF	Analog 2.5	Tx/Rx channel C positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 4 (pins not used in 10/100 Mbps modes).
PHY	P2_D3N	A7	I/O	ADIFF	Analog 2.5	Tx/Rx channel D negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 8 (pins not used in 10/100 Mbps modes).
PHY	P2_D3P	B7	I/O	ADIFF	Analog 2.5	Tx/Rx channel D positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 7 (pins not used in 10/100 Mbps modes).

PHY	P3_D0N	C4	I/O	ADIFF	Analog 2.5	Tx/Rx channel A negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 2.
PHY	P3_D0P	D4	I/O	ADIFF	Analog 2.5	Tx/Rx channel A positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the A data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 1.
PHY	P3_D1N	A4	I/O	ADIFF	Analog 2.5	Tx/Rx channel B negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 6.
PHY	P3_D1P	B4	I/O	ADIFF	Analog 2.5	Tx/Rx channel B positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the B data channel. In all three speeds, these pins generate the secondary side signal, normally connected to RJ-45 pin 3.
PHY	P3_D2N	C5	I/O	ADIFF	Analog 2.5	Tx/Rx channel C negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 5 (pins not used in 10/100 Mbps modes).

PHY	P3_D2P	D5	I/O	ADIFF	Analog 2.5	Tx/Rx channel C positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the C data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 4 (pins not used in 10/100 Mbps modes).
PHY	P3_D3N	A5	I/O	ADIFF	Analog 2.5	Tx/Rx channel D negative signal. Negative differential signal connected to the negative primary side of the transformer. This pin signal forms the negative signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 8 (pins not used in 10/100 Mbps modes).
PHY	P3_D3P	B5	I/O	ADIFF	Analog 2.5	Tx/Rx channel D positive signal. Positive differential signal connected to the positive primary side of the transformer. This pin signal forms the positive signal of the D data channel. In 1000-Mbps mode, these pins generate the secondary side signal, normally connected to RJ-45 pin 7 (pins not used in 10/100 Mbps modes).
POWER	VDD_1	F5	P			1.0 V power supply voltage for core.
POWER	VDD_10	P3	P			1.0 V power supply voltage for core.
POWER	VDD_11	P13	P			1.0 V power supply voltage for core.
POWER	VDD_12	N12	P			1.0 V power supply voltage for core.
POWER	VDD_13	N11	P			1.0 V power supply voltage for core.
POWER	VDD_14	M11	P			1.0 V power supply voltage for core.
POWER	VDD_15	L11	P			1.0 V power supply voltage for core.
POWER	VDD_16	K11	P			1.0 V power supply voltage for core.
POWER	VDD_17	J11	P			1.0 V power supply voltage for core.
POWER	VDD_18	H11	P			1.0 V power supply voltage for core.
POWER	VDD_19	G11	P			1.0 V power supply voltage for core.
POWER	VDD_2	G5	P			1.0 V power supply voltage for core.
POWER	VDD_20	F11	P			1.0 V power supply voltage for core.
POWER	VDD_3	H5	P			1.0 V power supply voltage for core.
POWER	VDD_4	J5	P			1.0 V power supply voltage for core.
POWER	VDD_5	K5	P			1.0 V power supply voltage for core.
POWER	VDD_6	L5	P			1.0 V power supply voltage for core.
POWER	VDD_7	M5	P			1.0 V power supply voltage for core.

POWER	VDD_8	N5	P			1.0 V power supply voltage for core.
POWER	VDD_9	N4	P			1.0 V power supply voltage for core.
POWER	VDD_A_1	P4	P			1.0 V power supply voltage for analog circuits.
POWER	VDD_A_2	P5	P			1.0 V power supply voltage for analog circuits.
POWER	VDD_A_3	P6	P			1.0 V power supply voltage for analog circuits.
POWER	VDD_A_4	P7	P			1.0 V power supply voltage for analog circuits.
POWER	VDD_A_5	P8	P			1.0 V power supply voltage for analog circuits.
POWER	VDD_A_6	P9	P			1.0 V power supply voltage for analog circuits.
POWER	VDD_A_7	P10	P			1.0 V power supply voltage for analog circuits.
POWER	VDD_A_8	P11	P			1.0 V power supply voltage for analog circuits.
POWER	VDD_A_9	P12	P			1.0 V power supply voltage for analog circuits.
POWER	VDD_AH_1	E5	P			2.5 V power supply for copper PHY interface.
POWER	VDD_AH_2	E6	P			2.5 V power supply for copper PHY interface.
POWER	VDD_AH_3	E7	P			2.5 V power supply for copper PHY interface.
POWER	VDD_AH_4	E8	P			2.5 V power supply for copper PHY interface.
POWER	VDD_AH_5	E9	P			2.5 V power supply for copper PHY interface.
POWER	VDD_AL_1	E10	P			1.0 V power supply for copper PHY analog circuits.
POWER	VDD_AL_2	E11	P			1.0 V power supply for copper PHY analog circuits.
POWER	VDD_IO_1	G4	P			2.5 V power supply for GPIO and miscellaneous I/Os.
POWER	VDD_IO_2	H4	P			2.5 V power supply for GPIO and miscellaneous I/Os.
POWER	VDD_IO_3	J4	P			2.5 V power supply for GPIO and miscellaneous I/Os.
POWER	VDD_IO_4	K4	P			2.5 V power supply for GPIO and miscellaneous I/Os.
POWER	VDD_IO_5	L4	P			2.5 V power supply for GPIO and miscellaneous I/Os.
POWER	VDD_IODDR_1	G12	P			1.35 V or 1.5 V power supply for DDR3/ DDR3L interface.

POWER	VDD_IODDR_2	H12	P			1.35 V or 1.5 V power supply for DDR3/ DDR3L interface.
POWER	VDD_IODDR_3	J12	P			1.35 V or 1.5 V power supply for DDR3/ DDR3L interface.
POWER	VDD_IODDR_4	K12	P			1.35 V or 1.5 V power supply for DDR3/ DDR3L interface.
POWER	VDD_IODDR_5	L12	P			1.35 V or 1.5 V power supply for DDR3/ DDR3L interface.
POWER	VDD_VS_1	N6	P			1.0 V or 1.2 V power supply voltage for SERDES1G and SERDES6G.
POWER	VDD_VS_2	N7	P			1.0 V or 1.2 V power supply voltage for SERDES1G and SERDES6G.
POWER	VDD_VS_3	N8	P			1.0 V or 1.2 V power supply voltage for SERDES1G and SERDES6G.
POWER	VDD_VS_4	N9	P			1.0 V or 1.2 V power supply voltage for SERDES1G and SERDES6G.
POWER	VDD_VS_5	N10	P			1.0 V or 1.2 V power supply voltage for SERDES1G and SERDES6G.
POWER	VSS_1	F6	P			Ground reference.
POWER	VSS_10	G10	P			Ground reference.
POWER	VSS_11	H6	P			Ground reference.
POWER	VSS_12	H7	P			Ground reference.
POWER	VSS_13	H8	P			Ground reference.
POWER	VSS_14	H9	P			Ground reference.
POWER	VSS_15	H10	P			Ground reference.
POWER	VSS_16	J6	P			Ground reference.
POWER	VSS_17	J7	P			Ground reference.
POWER	VSS_18	J8	P			Ground reference.
POWER	VSS_19	J9	P			Ground reference.
POWER	VSS_2	F7	P			Ground reference.
POWER	VSS_20	J10	P			Ground reference.
POWER	VSS_21	K6	P			Ground reference.
POWER	VSS_22	K7	P			Ground reference.
POWER	VSS_23	K8	P			Ground reference.
POWER	VSS_24	K9	P			Ground reference.
POWER	VSS_25	K10	P			Ground reference.
POWER	VSS_26	L6	P			Ground reference.
POWER	VSS_27	L7	P			Ground reference.
POWER	VSS_28	L8	P			Ground reference.
POWER	VSS_29	L9	P			Ground reference.
POWER	VSS_3	F8	P			Ground reference.
POWER	VSS_30	L10	P			Ground reference.
POWER	VSS_31	M6	P			Ground reference.
POWER	VSS_32	M7	P			Ground reference.

POWER	VSS_33	M8	P			Ground reference.
POWER	VSS_34	M9	P			Ground reference.
POWER	VSS_35	M10	P			Ground reference.
POWER	VSS_36	M4	P			Ground reference.
POWER	VSS_37	M12	P			Ground reference.
POWER	VSS_38	D8	P			Ground reference.
POWER	VSS_4	F9	P			Ground reference.
POWER	VSS_5	F10	P			Ground reference.
POWER	VSS_6	G6	P			Ground reference.
POWER	VSS_7	G7	P			Ground reference.
POWER	VSS_8	G8	P			Ground reference.
POWER	VSS_9	G9	P			Ground reference.
Reserved	RESERVED_0	F12	A	Analog	Analog 2.5	Leave floating/to testpoint
Reserved	RESERVED_1	E12	A	Analog	Analog 2.5	Leave floating/to testpoint
Reserved	RESERVED_3	C8	P	Analog		Tie to VSS
SERDES1G	S0_RXN	M1	I	TD	LVDS	Differential 1G data inputs/dual media.
SERDES1G	S0_RXP	M2	I	TD	LVDS	Differential 1G data inputs/dual media.
SERDES1G	S0_TXN	L2	O		LVDS	Differential 1G data outputs/dual media.
SERDES1G	S0_TXP	L1	O		LVDS	Differential 1G data outputs/dual media.
SERDES1G	S1_RXN	P1	I	TD	LVDS	Differential 1G data inputs/dual media.
SERDES1G	S1_RXP	P2	I	TD	LVDS	Differential 1G data inputs/dual media.
SERDES1G	S1_TXN	N2	O		LVDS	Differential 1G data outputs/dual media.
SERDES1G	S1_TXP	N1	O		LVDS	Differential 1G data outputs/dual media.
SERDES1G	S2_RXN	T2	I	TD	LVDS	Differential 1G data inputs/dual media.
SERDES1G	S2_RXP	R2	I	TD	LVDS	Differential 1G data inputs/dual media.
SERDES1G	S2_TXN	R1	O		LVDS	Differential 1G data outputs.
SERDES1G	S2_TXP	T1	O		LVDS	Differential 1G data outputs.
SERDES1G	S3_RXN	T4	I	TD	LVDS	Differential 1G data inputs/dual media.
SERDES1G	S3_RXP	R4	I	TD	LVDS	Differential 1G data inputs/dual media.
SERDES1G	S3_TXN	R3	O		LVDS	Differential 1G data outputs.
SERDES1G	S3_TXP	T3	O		LVDS	Differential 1G data outputs.
SERDES1G	S4_RXN	T6	I	TD	LVDS	Differential 1G data inputs.
SERDES1G	S4_RXP	R6	I	TD	LVDS	Differential 1G data inputs.
SERDES1G	S4_TXN	R5	O		LVDS	Differential 1G data outputs.
SERDES1G	S4_TXP	T5	O		LVDS	Differential 1G data outputs.
SERDES1G	S5_RXN	T8	I	TD	LVDS	Differential data inputs. 1G/2.5G.
SERDES1G	S5_RXP	R8	I	TD	LVDS	Differential data inputs. 1G/2.5G.
SERDES1G	S5_TXN	R7	O		LVDS	Differential data outputs. 1G/2.5G.
SERDES1G	S5_TXP	T7	O		LVDS	Differential data outputs. 1G/2.5G.
SERDES6G	RESERVED_39	R11	O	TD	CML	Leave floating

SERDES6G	RESERVED_40	T11	O	TD	CML	Leave floating
SERDES6G	RESERVED_41	T12	I		CML	Tie to VDDA
SERDES6G	RESERVED_42	R12	I		CML	Tie to VDDA
SERDES6G	S6_RXN	T10	I	TD	CML	Differential data inputs. 1G/2.5G/QSGMII.
SERDES6G	S6_RXP	R10	I	TD	CML	Differential data inputs. 1G/2.5G/QSGMII.
SERDES6G	S6_TXN	R9	O		CML	Differential data outputs. 1G/2.5G/QSGMII.
SERDES6G	S6_TXP	T9	O		CML	Differential data outputs. 1G/2.5G/QSGMII.
SERDES6G	S8_RXN	T14	I	TD	CML	Differential 1G data inputs.
SERDES6G	S8_RXP	R14	I	TD	CML	Differential 1G data inputs.
SERDES6G	S8_TXN	R13	O		CML	Differential data outputs. PCIe/2G5/1G
SERDES6G	S8_TXP	T13	O		CML	Differential data outputs. PCIe/2G5/1G
SI	SI_CLK	J1	I/O	ST, 3V	CMOS_2.5	Slave mode: Input receiving serial interface clock from external master. Master mode: Output driven with clock to external device. Boot mode: Output driven with clock to external serial memory device.
SI	SI_DI	H2	I	ST, 3V	CMOS_2.5	Slave mode: Input receiving serial interface data from external master. Master mode: Input data from external device. Boot mode: Input boot data from external serial memory device.
SI	SI_DO	H1	O	OZ, 3V	CMOS_2.5	Slave mode: Output transmitting serial interface data to external master. Master mode: Output data to external device. Boot mode: Output boot control data to external serial memory device.
SI	SI_nCS0	J2	I/O	ST, 3V	CMOS_2.5	Slave mode: Input used to enable SI slave interface. 0 = Enabled 1 = Disabled Master mode: Output driven low while accessing external device. Boot mode: Output driven low while booting from EEPROM or serial flash to internal VCore-III CPU system. Released when booting is completed.

8.0 PACKAGE INFORMATION

The VSC7513XKS is a is a lead-free (Pb-free), 256-pin, plastic ball grid array (BGA) with a 17 mm × 17 mm body size, 1 mm pin pitch, and 1.8 mm maximum height.

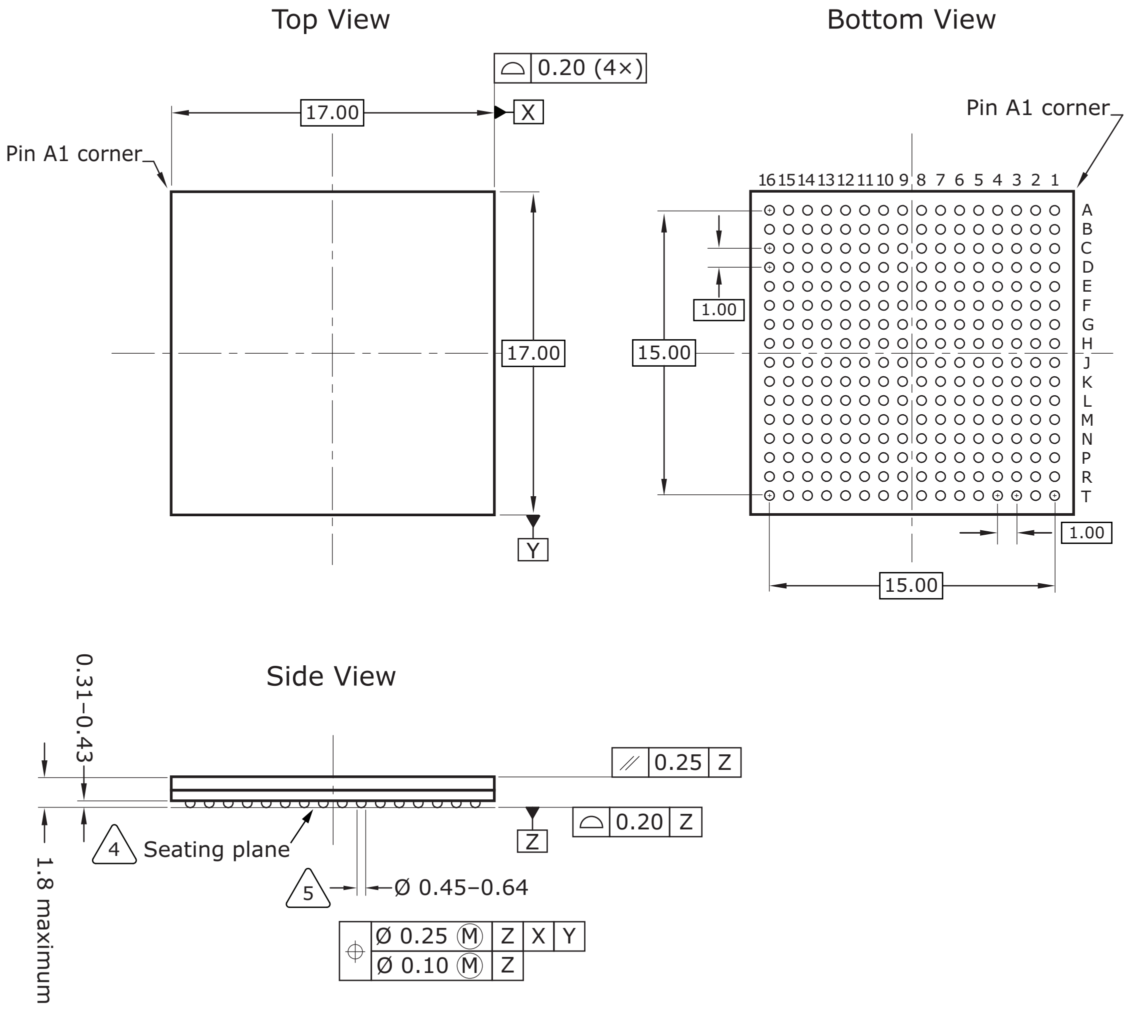
Lead-free products from Microchip comply with the temperatures and profiles defined in the joint IPC and JEDEC standard IPC/JEDEC J-STD-020. For more information, see the IPC and JEDEC standard.

This section provides the package drawing, thermal specifications, and moisture sensitivity rating for the VSC7513 device.

8.1 Package Drawing

The following illustration shows the package drawing for the VSC7513 device. The drawing contains the top view, bottom view, side view, detail views, dimensions, tolerances, and notes.

FIGURE 8-1: PACKAGE DRAWING



Notes

1. All dimensions and tolerances are in millimeters (mm).
2. Ball diameter is 0.50 mm.
3. Radial true position is represented by typical values.
4. Primary datum Z and seating plane are defined by the spherical crowns of the solder balls.
5. Dimension is measured at the maximum solder ball diameter, parallel to primary datum Z.

8.2 Thermal Specifications

Thermal specifications for this device are based on the JEDEC JESD51 family of documents. These documents are available on the JEDEC Web site at www.jedec.org. The thermal specifications are modeled using a four-layer test board with two signal layers, a power plane, and a ground plane (2s2p PCB). For more information about the thermal measurement method used for this device, see the JESD51-1 standard.

TABLE 8-1: THERMAL RESISTANCES

Symbol	°C/W	Parameter
θ_{JCtop}	2.43	Die junction to package case top
θ_{JB}	10.68	Die junction to printed circuit board
θ_{JA}	19.43	Die junction to ambient
θ_{JMA} at 1 m/s	16.25	Die junction to moving air measured at an air speed of 1 m/s
θ_{JMA} at 2 m/s	14.2	Die junction to moving air measured at an air speed of 2 m/s

To achieve results similar to the modeled thermal measurements, the guidelines for board design described in the JESD51 family of publications must be applied. For information about applications using BGA packages, see the following:

- JESD51-2A, *Integrated Circuits Thermal Test Method Environmental Conditions, Natural Convection (Still Air)*
- JESD51-6, *Integrated Circuit Thermal Test Method Environmental Conditions, Forced Convection (Moving Air)*
- JESD51-8, *Integrated Circuit Thermal Test Method Environmental Conditions, Junction-to-Board*
- JESD51-9, *Test Boards for Area Array Surface Mount Package Thermal Measurements*

8.3 Moisture Sensitivity

This device is rated moisture sensitivity level 4 as specified in the joint IPC and JEDEC standard IPC/JEDEC J-STD-020. For more information, see the IPC and JEDEC standard.

9.0 DESIGN GUIDELINES

This section provides information about design guidelines for the VSC7513 device.

9.1 Power Supplies

The following guidelines apply to designing power supplies for use with the VSC7513 device.

- Make at least one unbroken ground plane (GND).
- Use the power and ground plane combination as an effective power supply bypass capacitor. The capacitance is proportional to the area of the two planes and inversely proportional to the separation between the planes. Typical values with a 0.25 mm (0.01 inch) separation are 100 pF/in2. This capacitance is more effective than a capacitor of equivalent value, because the planes have no inductance or Equivalent Series Resistance (ESR).
- Do not cut up the power or ground planes in an effort to steer current paths. This usually produces more noise, not less. Furthermore, place vias and clearances in a configuration that maintains the integrity of the plane. Groups of vias spaced close together often overlap clearances. This can form a large slot in the plane. As a result, return currents are forced around the slot, which increases the loop area and EMI emissions. Signals should never be placed on a ground plane, because the resulting slot forces return currents around the slot.
- Vias connecting power planes to the supply and ground balls must be at least 0.25 mm (0.010 inch) in diameter, preferably with no thermal relief and plated closed with copper or solder. Use separate (or even multiple) vias for each supply and ground ball.

9.2 Power Supply Decoupling

Each power supply voltage should have both bulk and high-frequency decoupling capacitors. Recommended capacitors are as follows:

- For bulk decoupling, use 10 µF high capacity and low ESR capacitors or equivalent, distributed across the board.
- For high-frequency decoupling, use 0.1 µF high frequency (for example, X7R) ceramic capacitors placed on the side of the PCB closest to the plane being decoupled, and as close as possible to the power ball. A larger value in the same housing unit produces even better results.
- Use surface-mounted components for lower lead inductance and pad capacitance. Smaller form factor components are best (that is, 0402 is better than 0603).

9.2.1 REFERENCE CLOCK

The device reference clock can be a 25 MHz, 125 MHz, 156.25 MHz, or 250 MHz clock signal. It can be either a differential reference clock or a single-ended clock. For more information, see Section 6.1.2, Reference Clock Inputs.

9.2.2 SINGLE-ENDED REFCLK INPUT

An external resistor network is required to use a single-ended reference clock. The network limits the amplitude and adjusts the center of the swing. Depending on the clock driver output impedance, the resistor values may need to be adjusted so that the REFCLK_P signal meets requirements. For information about these requirements, see Table 6-2.

The following illustrations show configurations for a single-ended reference clock.

FIGURE 9-1: 2.5V CMOS SINGLE-ENDED REFCLK INPUT RESISTOR NETWORK

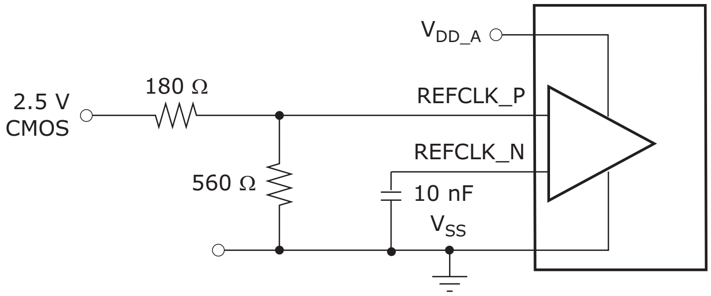
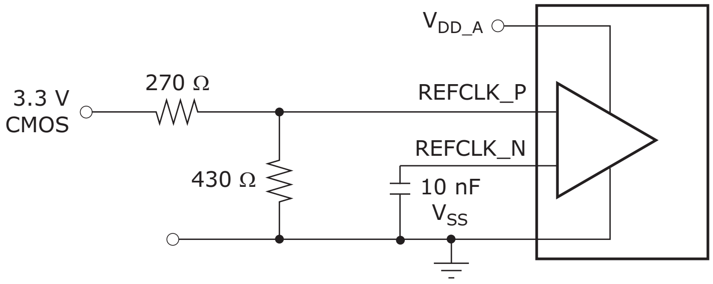


FIGURE 9-2: 3.3V CMOS SINGLE-ENDED REFCLK INPUT RESISTOR NETWORK



9.3 Interfaces

This section provides general recommendations for all interfaces and information related to the specific interfaces on the device.

9.3.1 GENERAL RECOMMENDATIONS

High-speed signals require excellent frequency and phase response up to the third harmonic. The best designs provide excellent frequency and phase response up to the seventh harmonic. The following recommendations can improve signal quality and minimize transmission distances:

- Keep traces as short as possible. Initial component placement should be considered very carefully.
- The impedance of the traces must match the impedance of the termination resistors, connectors, and cable. This reduces reflections due to impedance mismatches.
- Differential impedance must be maintained in a 100 Ω differential application. Routing two 50 Ω traces is not adequate. The two traces must be separated by enough distance to maintain differential impedance. When routing differential pairs, keep the two trace lengths identical. Differences in trace lengths translate directly into signal skew. Note that the differential impedance may be affected when separations occur.
- Keep differential pair traces on the same layer of the PCB to minimize impedance discontinuities. In other words, avoid using vias.
- Do not group all the passive components together. The pads of the components add capacitance to the traces. At the frequencies encountered, this can result in unwanted reductions in impedance. Use surface-mounted 0603 (or smaller) components to reduce this effect.
- Eliminate or reduce stub lengths.
- Reduce, if not eliminate, vias to minimize impedance discontinuities. Remember that vias and their clearance holes in the power/ground planes can cause impedance discontinuities in nearby signals. Keep vias away from other traces.
- Keep signal traces away from other signals that might capacitively couple noise into the signals. A good rule of thumb is to keep the traces apart by ten times the width of the trace.
- Do not route digital signals from other circuits across the area of the high-speed transmitter and receiver signals.
- Using grounded guard traces is typically not effective for improving signal quality. A ground plane is more effective. However, a common use of guard traces is to route them during the layout, but remove them prior to design completion. This has the benefit of enforcing keep-out areas around sensitive high-speed signals so that vias and other traces are not accidentally placed incorrectly.
- When signals in a differential pair are mismatched, the result is a common-mode current. In a well-designed system, common-mode currents should make up less than one percent of the total differential currents. Mode currents represent a primary source of EMI emissions. To reduce common-mode currents, route differential traces so that their lengths are the same. For example, a 5-mm (0.2-inch) length mismatch between differential signals having the rise and fall times of 200 ps results in the common-mode current being up to 18% of the differential current.

Note: Because of the high application frequency, proper care must be taken when choosing components (such as the termination resistors) in the designing of the layout of a printed circuit board. The use of surface-mount components is highly recommended to minimize parasitic inductance and lead length of the termination resistor.

Matching the impedance of the PCB traces, connectors, and balanced interconnect media is also highly recommended. Impedance variations along the entire interconnect path must be minimized, because these degrade the signal path and may cause reflections of the signal.

9.3.2 SERDES INTERFACES (SGMII, 2.5G, QSGMII)

The SGMII interface consists of a Tx and Rx differential pair operating at 1250 Mbps. The 2.5G interface consists of a Tx and Rx differential pair operating at 3125 Mbps. The QSGMII interface consists of a Tx and Rx differential pair operating at 5 Gbps.

The SerDes signals can be routed on any PCB trace layer with the following constraints:

- Tx output signals in a pair should have matched electrical lengths.
- Rx input signals in a pair should have matched electrical lengths.
- SerDes Tx and Rx pairs must be routed as 100 Ω differential traces with ground plane as reference.
- Keep differential pair traces on the same layer of the PCB to minimize impedance discontinuities. In other words, avoid the use of vias wherever possible.
- AC-coupling of Tx and Rx may be needed, depending on the attached PHY or module. External AC-coupling is recommended for use with most PHYs. SFP and SFP+ modules have internal AC-coupling, so they do not require additional AC-coupling capacitors. If AC-coupled, the VSC7513 SerDes inputs are self-biased. It is recommended to use small form factor capacitors, 0402 or smaller, to reduce the impedance mismatch caused by the capacitor pads.
- To reduce the crosstalk between pairs or other PCB lines, it is recommended that the spacing on each side of the pair be larger than four times the track width. The characteristic impedance of the pairs must predominantly be determined by the distance to the reference plane, and not the distance to neighboring traces.

9.3.3 SERIAL INTERFACE

If the serial CPU interface is not used, all input signals can be left floating.

The SI bus consists of the SI_CLK clock signal, the SI_DO and SI_DI data signals, and the SI_nCS0 device select signal.

When routing the SI_CLK signal, be sure to create clean edges. If the SI bus is connected to more than one slave device, route it in a daisy-chain configuration with no stubs. Terminate the SI_CLK signal properly to avoid reflections and double clocking.

If it is not possible (or desirable) to route the bus in a daisy-chain configuration, the SI_CLK signal should be buffered and routed in a star topology from the buffers. Each buffered clock should be terminated at its source.

9.3.4 PCI EXPRESS INTERFACE

The VSC7513 device does not accept spread spectrum modulated PCIe input. Although the device only supports PCI Express Base Specification Revision 1.1, the PCIe transmitter and receiver support the PCI Express Base Specification Revision 2.0 Electrical sub-block specifications under the following conditions:

- Only 2.5 gigatransfers per second (GT/s) is supported.
- Full swing output signaling is only supported when V_{DD_VS} = 1.2V.
- Low swing signaling is supported for V_{DD_VS} = 1.2V and V_{DD_VS} = 1.0V.

During PCIe startup (VCORE_CFG = 1001), only low swing signaling with no de-emphasis is supported, regardless of the V_{DD_VS} voltage. After startup, configure the PCIe interface as desired.

9.3.5 TWO-WIRE SERIAL INTERFACE

The two-wire serial interface is capable of suppressing small amplitude glitches. The duration of the glitch which can be suppressed is set in register ICPU_CFG::TWI_SPIKE_FILTER_CFG. The default value of 0 will suppress a 4ns glitch. It is recommended that SPIKE_FILTER_CFG be set to 12, which will suppress a 52ns glitch.

9.3.6 DDR3 SDRAM INTERFACE

The DDR3 SDRAM interface is designed to interface with 8-bit or 16-bit DDR SDRAM devices. The maximum amount of physical memory that can be addressed is one gigabyte. Possible combinations of memory modules are:

- One 8-bit device, connect to CS0 byte lane 0.
- Two 8-bit devices, connect to CS0 and both byte lanes.
- Four 8-bit devices, connect to both chip-selects and both byte lanes.
- One 16-bit device, connect to CS0 and both byte lanes.
- Two 16-bit devices, connect to both chip-selects and both byte lanes.

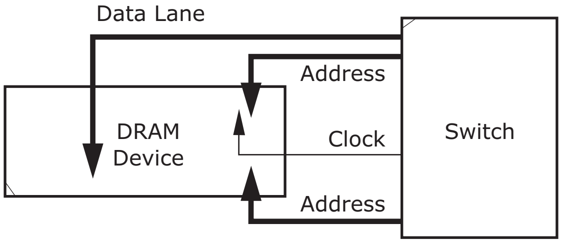
When using a single 8-bit device, the memory controller must be configured for 8-bit mode. All other configurations use 16-bit mode.

All signals on this interface must be connected one-to-one with the corresponding signals on the DDR SDRAM device. When using only one 8-bit device, the DDR_UDQS, DDR_UDQSn, DDR_UDM, and DDR_DQ[15:8] signals must be left unconnected. When using DDR2 SDRAM devices that has only four banks, the DDR_BA[2] signal must be left unconnected.

When using four 8-bit devices or two 16-bit devices, the CS1 memory modules must be placed in-between the VSC7513 device and the CS0 memory modules so that the DDR-DQS, DDR-DM, and DDR_DQ signals pass the CS1 devices first before reaching the CS0 devices.

The placement of the VSC7513 interface signals is optimized for point-to-point routing directly to a single DDR3 16-bit SDRAM.

FIGURE 9-3: 16-BIT DDR3 SDRAM POINT-TO-POINT ROUTING



Because reflections are absorbed by the devices, keep the physical distance of all the SDRAM interface signals as low as possible. Omit external discrete termination on the address, command, control, and clock lines.

When routing the DDR interface, attention must be paid to the skew, primary concern is skew within the byte lane between the differential strobe and the single-ended signals. Skew recommendations for the DDR interface are listed in the following table.

TABLE 9-1: RECOMMENDED SKEW BUDGET

Description	Signal	Maximum Skew
Skew within byte lane 0	DDR_LDQS/DDR_LDQSn	50 ps
Skew within byte lane 1	DDR_UDQS/DDR_UDQSn	50 ps
Skew within address, command, and control bus	DDR_CK/DDR_CKn DDR_nRAS DDR_CKE DDR_ODT[1:0] DDR_nCAS DDR_nWE DDR_BA[2:0] DDR_A[15:0]	100 ps
Skew between control bus clock and byte lane 0 clock	DDR_CK/DDR_CKn DDR_LDQS/DDR_LDQSn	1250 ps
Skew between control bus clock and byte lane 1 clock	DDR_CK/DDR_CKn DDR_UDQS/DDR_UDQSn	1250 ps
Control bus differential clock intra-pair skew	DDR_CK/DDR_CKn	5 ps

Power supply recommendations:

- Use a shared voltage reference between the VSC7513's device's DDR_Vref supply and the DDR device's reference voltage.
- Generate the DDR_Vref from the V_{DD_IODDR} supply using a resistor divider with value of 1 kΩ and an accuracy of 1% or better.
- Use a decoupling capacitance of at least 0.1 µF on the supply in a manner similar to V_{DD_IODDR} and V_{SS} to ensure tracking of supply variations; however, the time constant of the resistor divider and decoupling capacitance should not exceed the nRESET assertion time after power on.

- V_{DD_IODDR} pins must not share vias. Use at least one through for each V_{DD_IODDR} pin. The extra inductance from sharing vias may cause bit errors in the DDR interface.

Routing recommendations:

- DDR_CK/DDR_CKn must be routed as a differential pair with a 100 Ω differential characteristic impedance.
- DDR_*x*DQS/DDR_*x*DQSn must be routed as a differential pair with a 100 Ω differential characteristic impedance.
- To minimize crosstalk, the characteristic impedance of the single-ended signals should be determined predominantly by the distance to the reference plane and not the distance to the neighboring traces.
- The crosstalk should be below –20 dB.
- If the DDR interface is not used, connect V_{DD_IODDR} and DDR_V_{REF} to ground. Leave all other DDR signals unconnected (floating). V_{DD_IODDR} can also be left floating; however, DDR_V_{REF} must also then be left floating.

9.3.7 THERMAL DIODE EXTERNAL CONNECTION

The internal on-die thermal diode can be used with an external temperature monitor to easily and accurately measure the junction temperature of this device.

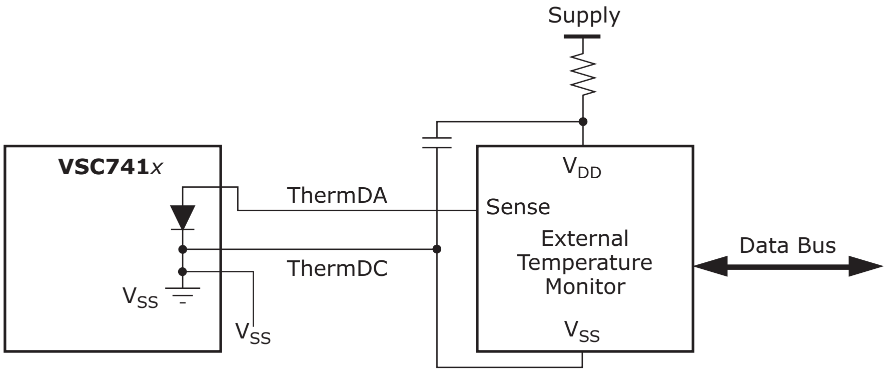
The thermal diode is extremely sensitive to noise. To minimize the temperature measurement errors, follow these guidelines:

- Route the THERMDC and THERMDA signals as a differential pair with a differential impedance less than 100 Ω.
- Place the external temperature monitor as close as is possible to this device.
- Add a 47 Ω resistor in series with the external temperature monitor supply to filter noise.

When using a grounded sensor circuit, place a de-coupling capacitor between the external temperature monitor supply pin and the THERMDC signal. For more information about using a grounded sensor circuit, see Figure 6-1.

Place the capacitor close to the external temperature sensor, as shown in the following illustration. Connect the external temperature monitor V_{SS} pin directly to the THERMDC pin, which has the connection to V_{SS}, as shown in the following illustration. Do not connect the external temperature monitor V_{SS} pin to the global V_{SS} plane.

FIGURE 9-4: EXTERNAL TEMPERATURE MONITOR CONNECTION



10.0 ORDERING INFORMATION

The VSC7513XKS is a is a lead-free (Pb-free), 256-pin, plastic ball grid array (BGA) with a 17 mm × 17 mm body size, 1 mm pin pitch, and 1.8 mm maximum height.

Lead-free products from Microchip comply with the temperatures and profiles defined in the joint IPC and JEDEC standard IPC/JEDEC J-STD-020. For more information, see the IPC and JEDEC standard.

The following table lists the ordering information for the VSC7513 device.

TABLE 10-1: ORDERING INFORMATION

Part Order Number	Description
VSC7513XKS	Lead-free (Pb-free), 256-pin, plastic ball grid array (BGA) with a 17 mm × 17 mm body size, 1 mm pin pitch, and 1.8 mm maximum height.

APPENDIX A: REVISION HISTORY

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the latest publication.

A.1 Revision A - 12/2022

The following is a summary of the changes implemented in the datasheet:

- The document was migrated to Microchip template.
- The document number was updated from VMDS-10490 to DS60001803A.
- The document has been updated to remove unwanted character (?) inserted during previous migration process.

A.2 Revision 4.2 - 05/2019

The following is a summary of the changes in revision 4.2 of this document.

- VeriPHY descriptions were updated. For functional details of the VeriPHY suite and operating instructions, see the *ENT-AN0125 PHY, Integrated PHY-Switch VeriPHY - Cable Diagnostics Application Note*.

A.3 Revision 4.1

Revision 4.1 of this datasheet was published in January 2018. The following is a summary of the changes in revision 4.1 of this document.

- Product overview was updated to reflect the correct number of supported 2.5G ports. For more information, see “Introduction”.
- The recovered clock settings table has been split into two. For more information, see Table 2-121 and Table 2-122.
- The SI timing master and slave mode specification sections were updated. For more information, see Section 6.2.8, SI Timing Master Mode Specifications and Section 6.2.9, SI Timing Slave Mode Specifications.
- The REFCLK rise and fall time was updated. For more information, see Table 6-11.

A.4 Revision 4.0

The following is a summary of the changes in revision 4.0 of this document.

- Product overview was updated to add feature highlights. For more information, see “General Features”.
- Information about configuring I/O delays was added. For more information, see Section 2.17.4, Configuring I/O Delays.
- Fan controller information was updated. For more information, see Section 3.8.10, Fan Controller.
- Electrical specifications were updated. For more information, see Section 6.0, Electrical Specifications.
- Pin type symbols were defined and the pins by function table was updated. For more information, see Table 6-36 and Section 7.2, Pins by Function.
- Moisture sensitivity level (MSL) is level 4.
- ESD (electrostatic discharge) was added. For human body model (HBM), it is a Class 2 rating. For charged device model (CDM), it is ±250 V.
- Information on design guidelines was added. For more information, see Section 9.0, Design Guidelines.
- Information on power supply sequencing was added. For more information, see Section 6.4.1, Power Supply Sequencing.

A.5 Revision 2.1

The following is a summary of the changes in revision 2.1 of this document.

- Current and power consumption values were updated. For more information, see Table 6-34.
- Limits for DDR3/DDR3L leakage (IL) changed to 40 uA. For more information, see Table 6-3 and Table 6-4.

A.6 Revision 2.0

Revision 2.0 was the first publication of this document.

THE MICROCHIP WEBSITE

Microchip provides online support via our WWW site at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website 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 website 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 website at: www.microchip.com/support

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.

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 https://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.

For information regarding Microchip’s Quality Management Systems, please visit www.microchip.com/quality.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, KoD, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, Trusted Time, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-6683-1727-3

MICROCHIP

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820