BCM88480

Packet Processing Architecture Specification

Design Guide

Table of Contents

Device Highlights

1 Introduction

The Broadcom® BCM88480, a seventh-generation StrataDNX™ switching device, is the industry's densest switching solution. The device, also known as Q2A, offers 800Gb/s network interfaces, deep-buffer traffic management, and up to 600G of ILKN I/F.

The BCM88480's packet processor offers a packet rate of 600 Mpps, large database scale, and best-in-class programmability and flexibility in processing functionality and database allocation.

This section provides a high-level summary of the BCM88480's Packet Processor functionality.

2 Interfaces

The BCM88480 has the following interfaces:

- Network interfaces (NIF):
  - 16 SerDes up to 53.125 Gb/s (PAM4)
  - 36 SerDes up to 25.78125 Gb/s
  - Port types supported include 1GbE, 10GbE, 25GbE, 40GbE, 50GbE, 100GbE, 200GbE, and 400GbE
  - 4 Interlaken (ILKN) interfaces
- Statistics interface:
  - Sharing network interface SerDes
  - Use Ethernet ports, up to 100GbE per statistics interface
  - Generate statistics records over packets
- External lookup interface:
  - Running over Interlaken interface, using NIF SerDes
- FlexE MACs of 5-400G
- External Lookup (ELK) Interface, using a NIF SerDes, enabling the direct connection of a Broadcom Knowledge-Base Processor (KBP) for database expansion.

3 Supported Protocols and Applications

The following is a partial list of protocols supported by the BCM88480 packet processor:

- Physical Layer: Ethernet Ver. 2, as well as IEEE 802.2 LLC and IEEE 802.2 SNAP; Support for packets without standard physical layer, via external framer
- Layer 2: IEEE 802.1Q, Metro Ethernet, Enterprise and Data Center, PPPoE
- Forwarding Protocols: Ethernet Bridging, IPv4/IPv6 Routing, IPv6 Segment Routing, MPLS, MPLS Segment Routing, VPLS (L2VPN), L3VPN, EVPN, IEEE 802.1ah (PBB, MAC-in-MAC), BIER, FCoE
- Termination and Encapsulation: IP Tunnels, VxLAN, NV-GRE, Geneve, Routing In/Out of Tunnels (RIOT)
- Test Features: Ethernet Performance (RFC2544), SAT (ITU Y.1564), TWAMP (RFC5357)
- OAM: Ethernet OAM (802.1AG and ITU-T Y.1731), BFD, RFC6374(LM/DM), and MPLS-TP OAM
- Instrumentation Applications: sFlow, Trajectory Trace, In-Packet Telemetry (IPT), Inband Network Telemetry (INT), Elephant Flow Detection

4 Databases and Object Scale

The BCM88480's on-chip databases are organized in a Modular Database (MDB), which supports the allocation of memory to individual databases according to device profiles. For example, in the L2-optimized profile (L2XL), large memory space is allocated to the Large Exact Match (LEM) physical database, where the MAC Table is stored. In the L3-optimized profile (L3XL), large memory space is allocated to the KAPS physical database, where IP FIB is stored.

The following table lists the various packet-processing objects in the BCM88480 in the Balanced MDB Profile.

Objects that are counted per-device increase their scale according to the number of devices in the system. For example, the Attachment Circuit (AC) scale is 128K per device and 256K in a two-device mesh.

Table 1: BCM88480 Object Scale (Balanced MDB profile)

Object	Balanced MDB Profile	Per
MAC Table	355K	—
MPLS-ILM	500K	—
IPv4 forwarding (subnet [+hosts])	750K[+355K]	—
IPv6 forwarding	375K	—
FEC Entries	115K	—
FEC ECMP groups	16K	—
FEC Protected Paths	128K	—
Attachment Circuit (AC)	64K	Dev
VLAN Translation	60K	Dev
Pseudo wire (PWE)	30K	Dev
Next-hop (ARP)	56K	Dev
Tunnels	30K	Dev
Router Interface (RIF)	16K	Sys
Multi points services (VSI)	16K	Sys
VSI to network VSI mapping	16K	Sys
Virtual Switch Instance (VSI)	16K	Sys
ACLs (TCAM)	24K	Dev

Table 1: BCM88480 Object Scale (Balanced MDB profile) (Continued)

Object	Balanced MDB Profile	Per
OAM maintenance points (MEP)	27K	Dev
OAM remote maintenance point (RMEP)	27.5K	Dev
Multicast	128K	Sys

The following tables demonstrate the BCM88480 database allocation flexibility facilitated by MDB. The profiles shown are the Balanced Profile, L2XL (Layer-2 optimized), L3XL (Layer-3 optimized), and the External-KBP, which assumes the IP FIB is stored on an external Broadcom Knowledge-Based Processor.

Different SKUs of the BCM88480 device offer different MDB sizes, scaling from basic applications to applications requiring large-scale databases, with maximum flexibility in database allocation.

Table 2: Select MDB Profiles (Maximum MDB SKU)

Object	Balanced	L2-XL	L3-XL
MAC Table (LEM)	355K	590K	77.5K
IPv4 forwarding subnets (LPM)	750K	32K	1.3M
IPv6 forwarding subnets (LPM)	375K	16K	650K
MPLS ILM (LEM)	500K	500K	150K
Attachment Circuit LIFs (InLIF1/2)	64K	64K	48K
ARP/Next-hop (EEDB)	72K	84K	88K
Multipoints services (VSI)	16K	64K	16K
FEC Entries (FEC)	115K	76.5K	256K

Table 3: Select MDB Profiles (Large MDB SKU)

Object	Balanced	L2-XL	L3-XL
MAC Table (LEM)	310K	388K	60K
IPv4 forwarding subnets (LPM)	386K	32K	650K
IPv6 forwarding subnets (LPM)	193K	16K	325K
MPLS ILM (LEM)	500K	500K	120K
Attachment Circuit LIFs (InLIF1/2)	32K	32K	32K
ARP/Next-hop (EEDB)	88K	96K	72K
Multipoints services (VSI)	16K	32K	16K
FEC Entries (FEC)	64K	76.8K	76.8K

Table 4: Select MDB Profiles (Basic MDB SKU)

Object	Balanced	L2-XL	L3-XL
MAC Table (LEM)	217K	294.5K	60K
IPv4 forwarding subnets (LPM)	386K	32K	650K
IPv6 forwarding subnets (LPM)	193K	16K	325K
MPLS ILM (LEM)	434K	500K	120K
Attachment Circuit LIFs (InLIF1/2)	32K	32K	32K

Table 4: Select MDB Profiles (Basic MDB SKU) (Continued)

Object	Balanced	L2-XL	L3-XL
ARP/Next-hop (EEDB)	68K	96K	80K
Multipoints services (VSI)	16K	32K	16K
FEC Entries (FEC)	64K	76.8K	76.8K

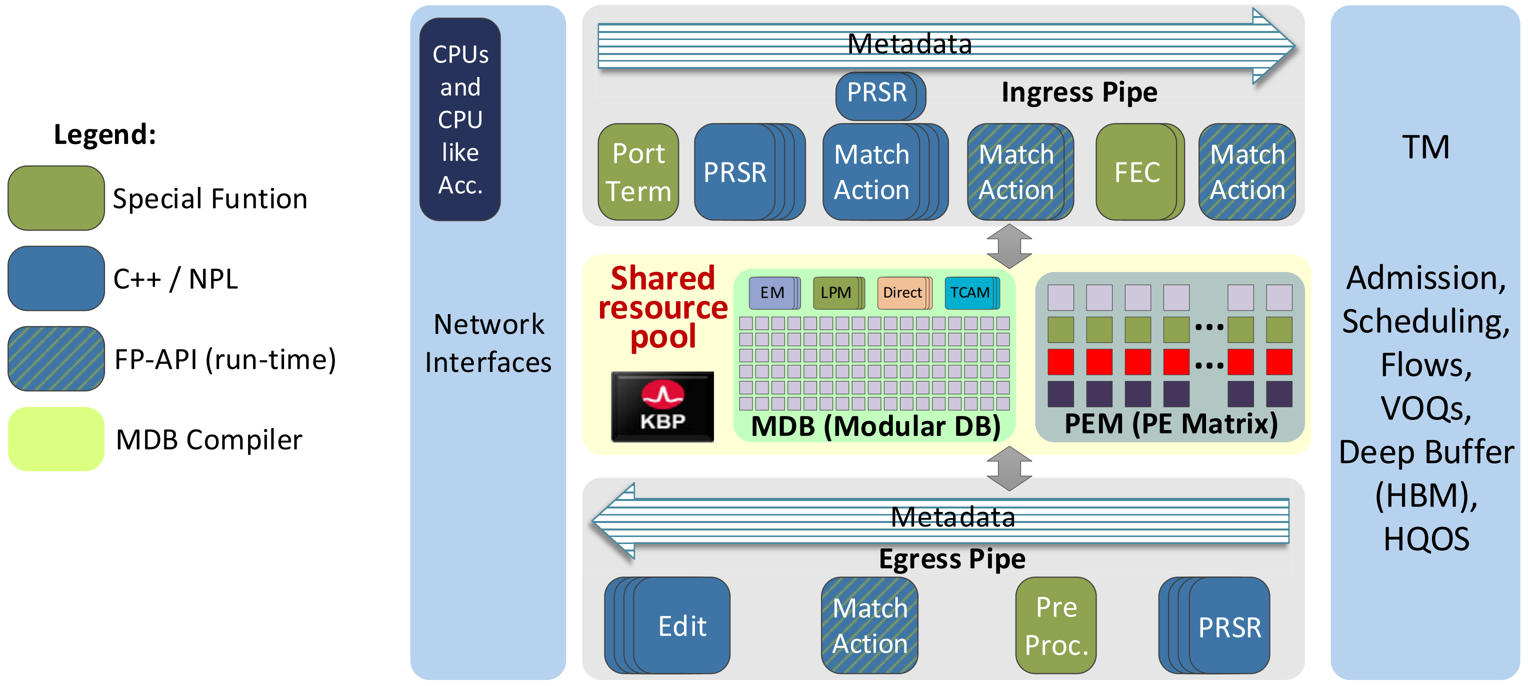
5 Programmability and Flexibility

The BCM88480 packet processor is highly programmable. Many hardware functions are controlled by a high-level object-oriented programming language, which is compiled into microcode controlling the hardware.

6 High-Level Block Diagram

The following figure provides a high-level view of the functional blocks and databases comprising the BCM88480 Packet Processor. The functionality of these blocks is described in detail in this document.

Figure 1: High-Level Block Diagram



Chapter 1: Centralized Switching Systems

1.1 Centralized Switch System

A Centralized Switch System comprises of two domains: A Packet Processing (PP) domain and a Traffic Management (TM) domain.

When traversing a Packet Processing domain, the packet experiences *single hop* processing, such as TTL decrement by one, VLAN association, and VLAN assignment. Forwarding plane operations are executed once: identification of the forwarding path for the packet by parsing the appropriate headers, assignment of a switching or forwarding context, and one or more lookups in the Forwarding Information Base (FIB).

Additional functions of the PP domain include assignment of TM properties, such as Class of Service (CoS) and Drop Precedence (DP).

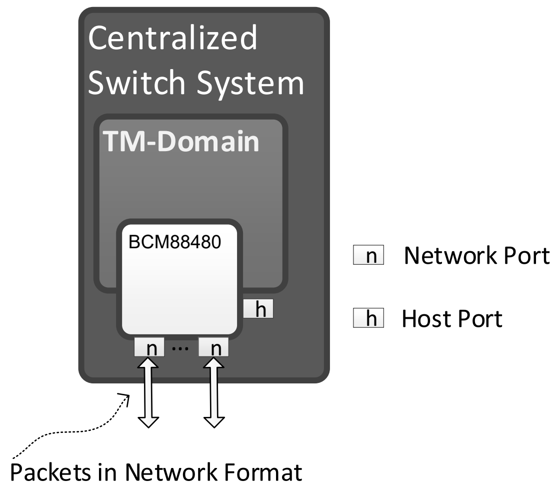
When traversing a Traffic Management domain the packet experiences a single scheduling-hop and single enqueue/ dequeue from the main packet buffer.

The TM domain executes buffering and scheduling for various traffic classes according to the Quality of Service (QoS) policy. Additional TM functions include congestion management, metering, shaping, and differentiated service.

A StrataDNX Centralized Switch System can be constructed of a single device (for example, one BCM88480). The Centralized Switch System is considered a single PP domain and a single TM domain.

The figure below provides a logical view of an example Centralized Switch System.

Figure 2: Centralized Switch System—Logical View



Chapter 2: Device Organization Overview

2.1 Introduction

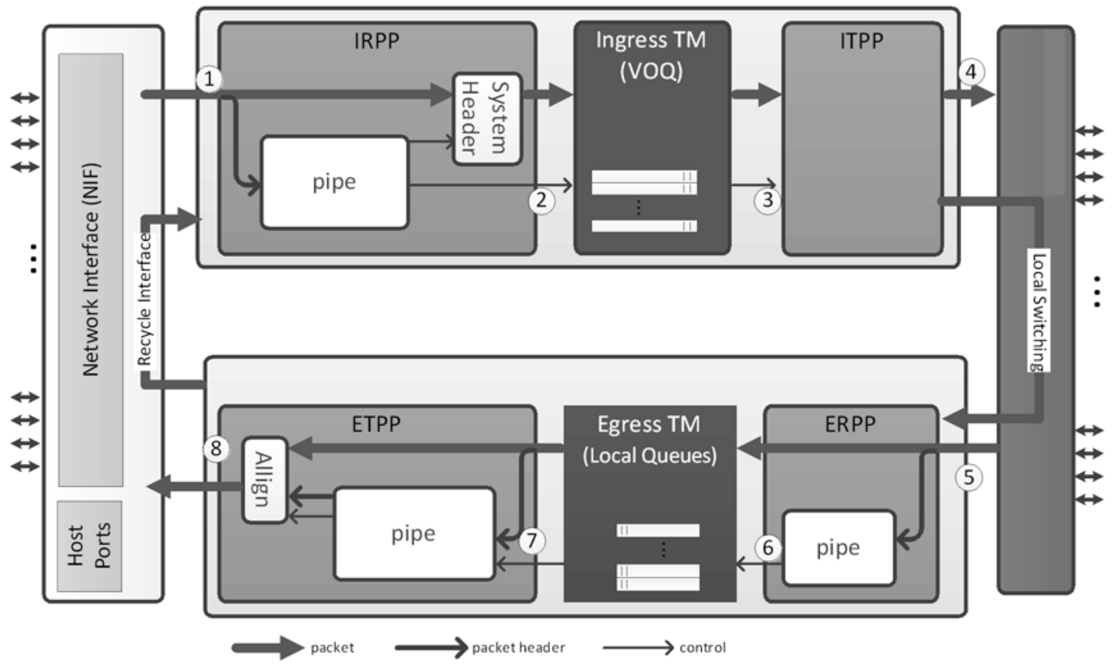
The StrataDNX architecture provides a system-level traffic-management and packet-processing solution.

In this section, we describe the components of the BCM88480 device and their organization in the ingress and the egress pipes. This is done by describing the processing performed on a packet from the point it enters the system, through all Packet Processing and Traffic Management stages, to the point it exits the system.

2.2 System Packet Walk

The diagram below illustrates the flow of packets through a system comprised of one TM domain (scalable switch system). The packet-processing functions (light gray rectangles) are shown in relation to the StrataDNX traffic management and fabric functions.

Figure 3: Packet Flow in the StrataDNX Architecture (Scalable Switch System)



Traffic that crosses TM domains (that is, traverses a stacking port in a distributed switch system) is subject to multiple stages of ingress processing followed by egress processing. However, as noted previously, packet processing, as shown in the diagram is executed only once, in the ingress pipe of the first TM domain. In subsequent TM domains, link-layer or network-layer processing is not performed, only switching across TM domains according to the decision made by the initial Ingress processing.

2.2.1 Ingress Pipe

- RX Interfaces: Packets enter the ingress pipeline through one of the following:
  - Network interfaces
  - External/internal host interface such as CPU, OAMP, SAT, OLP, etc.
  - Recycling (loopback) interface, from the egress pipe on the same device
- Ingress Receive PP (IRPP): Parses and classifies the packet, and performs lookups in various tables. Based on lookup results, IRPP determines the packet's egress port and editing instructions, and generates enqueue requests to the ingress TM (ITM) for each packet. The IRPP also encapsulates the packet by a system header, carrying information required for egress processing.
  The processing at the IRPP may include network-layer headers processing, and/or custom headers processing, such as Ingress-TM Header (ITMH). ITMH is typically used when an external Network Processing Unit (NPU) performs the network-layer processing, and appends the ITMH that includes information used by IRPP to generate the enqueue requests.
- Ingress TM (ITM): A traffic manager that manages tens of thousands of queues in On-chip Buffers (OCB) and optionally in large-capacity buffers, located in an off-chip High Bandwidth Memory (HBM). Queues are flexibly allocated to functions, such as a Virtual Output Queue (VOQs), and Egress-Flow Queues.
  ITM and its queues are part of the end-to-end Traffic Manager service, which includes, among other functions such as shaping, scheduling, and congestion management. ITM also executes packet replication for multicast, mirroring, and snooping, as well as packet discard.
  According to the enqueue request, ITM queues the packet in a VOQ associated with the packet's destination port(s).
- Ingress Transmit PP (ITPP): The ingress TM stores a single copy of the packet, regardless of the number of copies that are generated at the ITM processing stage. ITPP edits various fields in the system header, separately for each copy, when it is de-queued from its VOQ.
  Among others, ITPP post-dequeue system header editing includes stamping the packet size, destination port, and Copy-Unique-Data (CUD) fields for copies of packets that are ingress replicated.
  ITPP also performs ingress latency measurements, measured from the time the packet is received to the time it is dequeued from the VOQ.

2.2.2 Egress Pipe

- Egress Receive PP (ERPP): Decodes system headers and, based on the extracted information, generates enqueue requests to the Egress TM.
  Additionally, egress ACLS and various other filters are applied, such as source-port filtering (multicast pruning), STP, VSI/VLAN membership, Split-Horizon, and TTL Scoping.
  For multicast packets, a single packet may be replicated to multiple ports. Each copy is stamped by a Copy-Unique data field, which is later used by the ETPP to allow unique processing of the packet copy.
- Egress TM (ETM): A local traffic manager that manages class-of-service queues for destination ports (OTM-Port).
- Egress Transmit PP (ETPP): Edits the packet before it leaves the device, using information encoded in the system headers, in local databases, and in the packet's own network headers. Specifically, ETPP performs the following:
  - Removes the system header, and any network headers that the ingress PP indicates as terminated
  - Edits the network layer header used for the forwarding decision, such as Ethernet, MPLS, or IP
  - Encapsulates the packet by constructing and appending network and/or link-layer headers or, possibly, a custom header, such as Outgoing TM Header (OTMH)
- TX Interfaces: Packets exit the egress pipeline through one of the following interfaces:
  - Network interfaces
  - External/internal host interface such as CPU, OAMP, SAT, OLP, etc.
  - Recycling (loopback) interface, from the egress pipe on the same device

2.3 Internal Hosts

In addition to the packet processing and traffic management pipes, the BCM88480 also includes internal sources of traffic used for management, testing and various hardware acceleration functions.

Each of these blocks appears as a source port on the ingress pipe and a destination port on the egress pipe. These blocks are described briefly below.

2.3.1 CPU Management Interface Controller (CMIC)

The CPU Management Interface and Control (CMIC) belongs to the iProc subsystem. The CMIC and iProc include a DMA-based bridge between the BCM88480 packet interface and the PCI Express (PCIe) interface to the CPU. The CMIC may be used by the CPU to inject packets to the ingress pipe and to receive packets from the egress pipe.

In addition, the iProc core includes embedded ARM processors and interfaces to peripheral devices such as GPIO, SPI, UART, etc.

2.3.2 Service Activation Test (SAT)

The Service Activation Test (SAT) engine is a traffic generator and analyzer whose goal is to support testing protocols such as Service Activation Testing (RFC2544, ITU Y.1564, ITU Y.1553) and Two-Way Active Management Protocol (TWAMP, RFC5357).

2.3.3 OAM Processor (OAMP)

The Operations, Administration, and Management Processor (OAMP) is a hardware acceleration engine in charge of generating and processing OAM packets and executing OAM functions.

OAMP capabilities include:

- Generating and receiving Continuity Check Message (CCM) packets, maintaining the status of remote MEPs, and generating events upon status changes.
- Generating and receiving Loss Measurement (LM) and Delay Measurement (DM) messages, and calculating the link performance accordingly.

OAMP’s functionality complements the identification and classification functionality of OAM packets, which is implemented in the packet processor pipe (IRPP and ETPP). The full list of protocols supported by the OAMP and lower-level details of its operation are provided in the relevant sections.

2.3.4 MACT Learning Off-Load Processor (OLP)

The Off-Load Processor (OLP) is a part of the MACT Learning subsystem, responsible for communicating with OLPs in other devices to distribute MACT Learn events.

On the transmission direction, OLPs aggregate learn events from local MACT management, construct packets, and inject the packets into the ingress pipe for transmission to other devices. On the receive direction, OLPs receive packets containing learn events from OLPs in other devices via the egress pipe, parses the packet, and passes the events to the local MACT management.

Chapter 3: Ports and Interfaces

3.1 Introduction

StrataDNX BCM88480 system architecture distinguishes between system ports, which are global in their contexts and are used in the entire system (PP and TM), and Packet Processing Ports (PP-Ports) whose contexts are local to a single device PP, and are only used in the PP.

A port may be a network port, a channel of a network port, an aggregated port (LAG), an internal source of traffic (for example, CPU) or another type of port. In this chapter, we describe the different types of ports and interfaces in the StrataDNX system, and their characteristics.

3.2 System Interfaces

3.2.1 Overview

Packets are received and transmitted by the BCM88480 device through physical interfaces. At the entry point to the Ingress PP, the Physical Interface ID and the Channel ID (if exists) are mapped into ports.

Specifically, the physical interface and channel are mapped to:

- Reassembly Context, up to 256 contexts are available to the core.
- Port Termination Context (PTC), which is mapped in IRPP to Source-System-Port (SSP) and In-PP-Port, in addition to other attributes required for starting the parsing of the packet.

In many cases the PTC and the Reassembly Context are the same entity. That is, a non-channelized interface will be allocated one PTC and Reassembly Context. An interleaved channelized interface will be allocated one PTC and Reassembly Context for each channel. However, channelized which are non-interleaved may be allocated a PTC for each channel, and one Reassembly Context for the entire interface.

The types of interfaces existing in the system are explained in the following subsections.

3.2.2 Network Interface

Network interfaces are connected to physical networking ports such as an Ethernet PHY or OTN Framer. Some network interfaces are non-channelized (for example, 10G Ethernet or 100G Ethernet) and others are channelized interfaces (for example, Interlaken).

3.2.3 Recycle Interface

The Recycle Interface connects the egress pipe output to the ingress pipe input. It is used in cases where packets need further processing. The Recycle Interface is channelized, such that it may be shared among multiple recycle ports.

The traffic on the Recycle Interface can be classified to two main groups:

- Scheduled Traffic: some applications require 2-pass processing. Traffic belonging to these applications is directed in IRPP to one of the recycle ports. This traffic is queued in a VOQ associated with the port and is scheduled using the system's TM mechanisms.
- Redirected Traffic: packets may be caught by egress ACLs or egress pipe filters, or outbound-mirrored, at a point where it is too late to redirect them to a new port. This traffic is redirected to the Recycle Interface to undergo further processing in IRPP. This traffic is not scheduled, and theoretically may be the entire traffic on the egress pipe.

The traffic on the Recycle Interface is channelized and may be interleaved, so a Reassembly Context and a PTC are associated with each channel. Traffic on the Recycle Interface is directed from the egress pipe on the core to the ingress pipe on the core.

Packets on the Recycle Interface are accompanied by a Recycle Command which translates to parsing context and initial Packet Actions in IRPP. In addition, packets are typically encapsulated by Recycling Header (RCH) carrying additional information to assist the 2nd-pass processing

3.2.4 Interfaces to Internal Hosts

Internal Interfaces are interfaces through which packets are received from or transmitted to internal hosts such as a local CPU (via CMIC, or directly), OAMP engine, SAT engine, OLP, etc.

3.3 Ports

When a packet is processed in the BCM88480, ports are associated with it, and are used in processing the packets. There are two main types of ports in the system:

- System Ports, which are global entities. A system port number is unique to a port in the entire system. Up to 32,768 system ports are supported in the StrataDNX BCM88480 system.
- Packet Processing Ports (PP Ports) are local entities used for processing within a PP core in a single device. Up to 256 PP Ports are supported in the PP core.

Mapping between System Ports (global) and PP Ports (local) is done in a number of points in the system, as described in Section 3.4, Port Mapping in the Packet Processor.

Figure 8 illustrates a Distributed Switch System (DSS) comprised of multiple Traffic Management (TM) domains. The diagram shows the different types of system ports in the DSS.

3.3.1 System Ports

A System Port (SP) is identified by a system-level (that is, global) unique number, and connects any entity that generates or terminates packets to the switch pipe.

- An SP often corresponds to a physical interface but may also correspond to a logical interface such as a Virtual Machine (VM) in a data center, or a GEMID in PON.
- Every packet that enters the system is associated with a Source System Port (SSP). This association stays with the packet until it is transmitted out of the system.
- Every packet that is transmitted out of a system through its Destination System Port (DSP).

3.3.2 System Ports Aggregate (SPA)

A number of system ports may be grouped, in Layer 2, into one logical port known as a Link Aggregation Group (LAG). In that case, the system ports are members of a System Port Aggregate (SPA). An SPA has the following characteristics:

- A SPA is a collection of System Ports (SPs).
  - An SP may be a member of only one SPA.
  - It is common for a SPA to have a single SP member.
- When a packet enters the system via a LAG, it is associated with a Source SPA (SSPA). This association stays with the packet until it is sent out of the system.
- Before a packet is transmitted out of the system via a LAG, it is assigned a Destination SPA (DSPA)
  - To transmit the packet, an actual member of the DSPA is selected using a Load Balancing Key (LB-Key), in a procedure called LAG Resolution.

3.3.3 Stacking Port/Stacking Port Aggregate

Stacking ports are used for connecting between TM-Domains in a distributed system switch. Stacking ports have the following characteristics:

- Stacking Ports commonly use Link Aggregation.
- Packets transmitted across Stacking Ports are encapsulated by a system header, with a stacking extension.
- Special PP procedures are applied to packets received through stacking ports to avoid loops in the stacking system.

The table below provides a summary of the different types of system ports and their system function.

Table 5: System Port Descriptions

Port Acronym	Port Name	Description
SP	System Port	Object that receives or transmits a frame at the DSS boundary, for example, network port (100GE), channelized network port (CoE, Interlaken).
SSP/DSP	Source/Destination System Port	System port from which a frame has entered/is set to leave the DSS.
SPA	System Port Aggregate	Aggregate Port at DSS level. In L2, this is a LAG.
SSPA/DSPA	Source System Port Aggregate	Aggregate Port at DSS input/output.
	Stacking Port	A TM-Domain port, through which inter-TM-Domain traffic is transferred.
	Stacking Port Aggregate	Aggregate (LAG) stacking port.

3.3.4 Traffic Management Ports (TM Ports)

Traffic management ports are used by the TM in order to perform packet reassembly.

3.3.4.1 Input TM (ITM) Port

As a packet enters a TM Domain, the network interface assigns to it an ITM Port, mapped from the incoming Physical Interface ID and Channel ID (if applicable). The ITM Port is associated with a Traffic Manager reassembly context.

The recycling interface and internal interfaces are treated as physical interfaces on the ingress TM.

3.3.4.2 Output TM (OTM) Port

Before a packet is transmitted out of a TM Domain, it is assigned an OTM Port. The OTM Port is associated with an egress queuing structure and corresponding flow-control reaction points.

Packets from VOQs are scheduled into an OTM Port according to a configurable scheduling hierarchy.

Upon transmission, the packet's OTM Port is mapped to outgoing interfaces and channels.

3.3.5 Packet Processing Ports (PP Ports)

PP Ports are ports of local scope in the packet processor in which they are used:

- PP Source System Port (PP-SSP), also known as In-PP-Port, is a local representation of SSP/SSPA
- PP Destination System Port (PP-DSP), also known as Out-PP-Port, is a local representation of DSP/DSPA

The PP Ports are summarized in the table below.

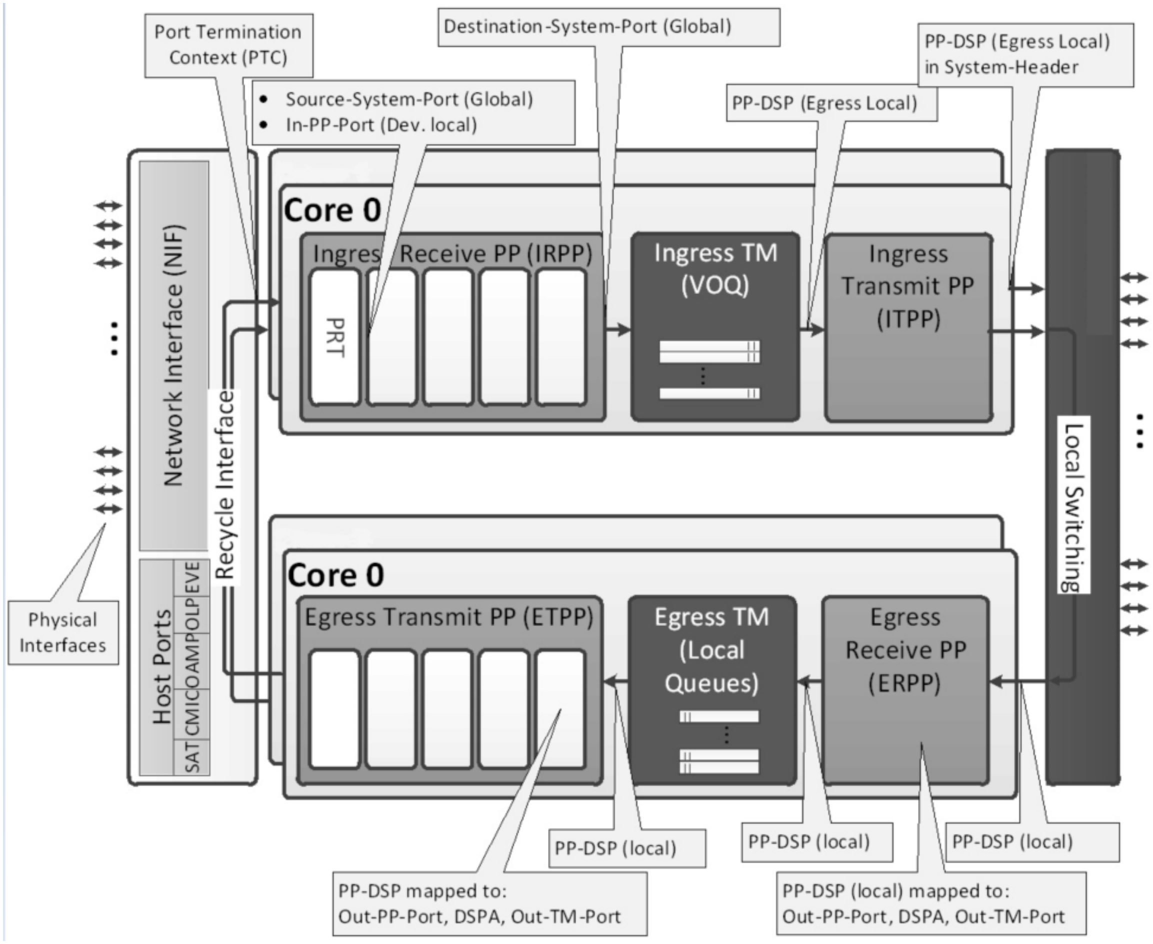
Table 6: PP Port Descriptions

Port Acronym	Port Name	Description
PP-SSP	Source port in the local PP context	The local port on which a frame has arrived into the system. The port may represent a LAG port. Also known as In-PP-Port.
PP-DSP	Destination port in the local PP context	The local port on which a frame is set to leave the system. The port may represent a LAG port. Also known as Out-PP-Port.

3.4 Port Mapping in the Packet Processor

In this subsection, we described how the port-related attributes are assigned to packets and are used in the packet processor, as illustrated in Figure 4.

Figure 4: Ports and Interfaces Handling


Port Termination Context (PTC)
Destination-System-Port (Global)
PP-DSP (Egress Local) in System-Header
• Source-System-Port (Global)
• In-PP-Port (Dev. local)
PP-DSP (Egress Local)
Core 0
Ingress Receive PP (IRPP)
PRT
Ingress TM (VOQ)
Ingress Transmit PP (ITPP)
Network Interface (NIF)
Recycle Interface
Local Switching
Host Ports
SAT
CMIC
OAMP
OLP
EVE
Physical Interfaces
Core 0
Egress Transmit PP (ETPP)
Egress TM (Local Queues)
Egress Receive PP (ERPP)
PP-DSP (local)
PP-DSP (local)
PP-DSP (local)
PP-DSP mapped to: Out-PP-Port, DSPA, Out-TM-Port
PP-DSP (local) mapped to: Out-PP-Port, DSPA, Out-TM-Port

With reference to Figure 4, the packet-processing of port-related attributes is described below.

3.4.1 Ingress Device

- Physical Interface (Network, Recycle, or Internal): receives the packet and assigns to it a Port Termination Context (PTC).
- Port Termination Block (PRT): maps the PTC to Source System Port (SSP) and In-PP-Port, as well as other values needed to start parsing and processing the packet.
- IRPP: uses In-PP-Port in identifying the packet's Input Logical Interface (InLIF).
- PMF: block may use the SSP and In-PP-Port as ACL criteria.
- The forwarding result, resolved in IRPP, is normally a Destination System Port (DSP), which is passed to ITM as a field in the TM-Command.
- ITM: translates the DSP, a global value, to a PP-DSP, which is local in the context of the egress device. Before that, if LAG resolution is required, ITM selects a LAG member and updates the DSPA accordingly.
- ITPP: updates the writes the PP-DSP into the system header (FTMH), to be used by the Egress device.

3.4.2 Egress Device

- ERPP: maps the PP-DSP from FTMH to the Destination System Port (DSP) and Out-TM Port.
- Egress PMF: may use the DSP, which is a global value, and/or PP-DSP, which is local to the egress device context as ACL criteria.
- ERPP: may redirect the packet to a new destination by updating the PP-DSP and Out-TM Port, according to ACLs or filter results.
- ERPP: passes the PP-DSP and Out-TM Port (updated or not) to ETM.
- ETPP: maps the PP-DSP received from ETM to the Destination System Port (DSP) and uses the DSP and PP-DSP in filters and processing functions.
  ETPP may not change the destination port of the packet.
- Physical Interface (Network, Recycle, or Host): the packet is transmitted via the physical interface designated in its Out-TM Port.

Chapter 4: System Headers

4.1 System Headers Overview

The system headers are a set of headers that encapsulate the packets as they travel in the system. They are comprised of mandatory headers, and optional headers and extensions, which may be added according to system configuration or packet types.

The primary goal of the system header is to pass packet processing results, and additional information, from the ingress pipe to the egress pipe, whether they reside on different devices, or on the same device.

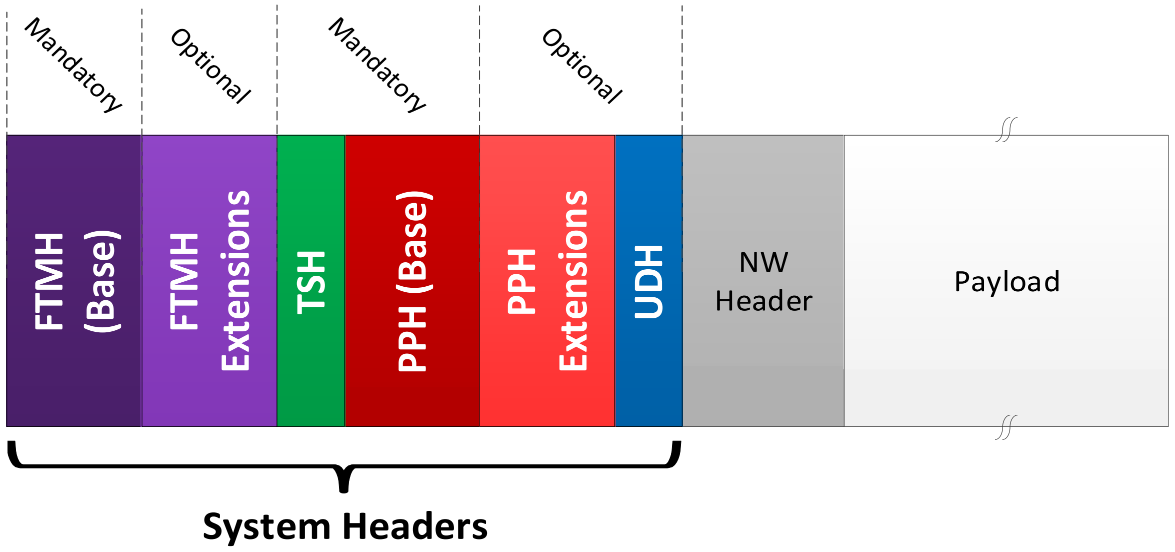
Some properties of the system headers:

- Initial Construction: System headers are initially constructed once the Ingress Packet Processing is completed in IRPP.
- Updates: System headers may be updated in specific locations along the packet's path, most notably in ITPP.
- Removal: System headers are removed before the packet is transmitted outside the system via a network port.
- Stacking Headers: In a distributed system, when the packets are transmitted via a stacking port, the system headers are not removed.
- Injection: When packets are injected into the system, for example, by CPU, it may be already encapsulated by a system header.
- External Packet Processing: In systems where packet processing is done externally, the results of the external processing are encoded in a system header, which encapsulates the packet when it enters the system.

4.2 System Header Structure

The system header is comprised of mandatory headers and optional headers, as illustrated in Figure 5 and explained in detail in the following subsections.

Figure 5: System Header Structure



4.2.1 Fabric Traffic Management Header

The Fabric Traffic Management Header (FTMH) is comprised of a base header, which is always present, and extensions, which are optional. The extensions may be added according to the packet type (for example, BIER Extension) or according to the system configuration (for example, Stacking Extension).

The FTMH base header size is 80 bits, and includes the following information:

- Destination Information: Destination Port, in egress device local context (PP-DSP) and Outgoing Logical Interface (OutLIF), (unicast)
- Multicast-ID or Multicast-Replication Index (multicast packets)
- Source System Port (SSP/SSPA): Where the packet entered the switch system
- QoS Parameters: Traffic Class (TC), Drop Precedence (DP), Congestion Indicators (CNI, ECN)
- Packet Size
- etc.

More information on FTMH can be found in the BCM88480 TM Architecture Specification.

4.2.1.1 FTMH Extensions

FTMH may be accompanied by zero or more extensions, carrying various types of information required in the system. The addition of extensions is dependent upon the type of the packet or system configuration. Some FTMH extension examples are listed below:

- Load Balancing Key (LB-Key) Extension: Carrying LB-Keys calculated in IRPP to be used in LAG resolution and in Network QoS stamping in the egress pipe.
- FTMH Stacking Extension: Carrying the stacking route history, to avoid loops in the distributed switch system, as well as traffic class determined by the original ingress pipe.
- FTMH Application Specific Extension (ASE): Associated with a number of applications, such as OAM, 1588v2, Trajectory Trace, Inband Network Telemetry (INT), and ERSPAN.

4.2.2 Timestamp Header

Timestamp Header (TSH) carries the RX Timestamp, recording the time the packet entered the system, generated by the RX MAC. The timestamp is encoded into the TSH when the system header is created.

4.2.3 Packet Processing Header

Packet Processing Header (PPH) is also comprised of a base header, which is always present, and zero or more extensions. The extensions may be added according to the packet type or according to system configuration.

PPH carries the information required by the Egress Packet Processing Pipe to edit the outgoing packet's header (namely, termination of existing headers and construction of new encapsulation headers) and forward the packet.

4.2.4 User Defined Header

The system builder can configure BCM88480 device build a User Defined Header (UDH), carrying any content from the ingress device to the egress device. UDH presence and its size may be different for each packet, as is indicated by the UDH-Base field, which is always present. UDH-Base may indicate that there is no UDH, or it may provide information on UDH structure and size.

4.2.5 ITM and PTC Headers

When a packet is injected to the system by an internal host (for example, CPU or OAMP), or by an external processing device (for example, NPU or Framer) it should be encapsulated by one of the headers listed in this subsection. More information on incoming headers can be found in the BCM88480 TM Architecture Specification.

4.2.6 Port Termination Control Header

Port Termination Control Header (PTCH) enables a packet source (for example, internal or external host) to control the port termination stage processing. This is useful to enable a packet source to set the In-PP-Port and SSPA attributes to present an injected packet as a packet from a desired source port.

PTCH existence is indicated by a configuration of the Port Termination Context (PTC), a mapping of the source interface and channel. In other words, a source port may be configured to expect all packets received through it to be encapsulated by PTCH.

4.2.7 Input Traffic Management Header

Input Traffic Management Header (ITMH) is used when the entire packet processing functionality is bypassed, (for example, when packet processing is executed by an external NPU). The NPU encapsulates the packet with ITMH and passes it to the BCM88480 network port.

ITMH encodes the packet’s destination, as resolved by the external processor and a Sniff Profile, which is mapped later, by ITPP, to a full system header, passed to the egress pipe.

Chapter 5: Packet Actions

5.1 Introduction

As packets travel through the BCM88480 Packet Processing pipe, up to four packet actions may be attached to each packet. The packet actions are attributes that travel with the packet along the pipe. They are used as an efficient method to trigger the generation of packet copies and for modifying packet attributes (for example, destination) on the main packet and the copies.

The main packet action is the *Forward Action*, encoding how the main copy of the packet is processed. Normally, the Forward Action indicates forwarding the packet to its destination, as determined by forwarding lookup. In cases when a packet is trapped, the Forward Action will encode trapping.

In addition, up to three copies may be generated by attaching specific Action-Profiles:

- Snoop Copy (Snoop-Action-Profile): targeted to a local or remote CPU
- Mirror Copy (Mirror-Action-Profile): targeted to a monitoring host (local or remote)
- Statistical Sampling Copy (Statistical-Sampling-Action-Profile): targeted to an instrumentation statistics collector (local or remote)

The packet Action-Profiles are used as an efficient method for specifying a certain set of attributes to each packet copy. Action-Profiles may be updated in various points in the pipe, using strength-based resolution where conflicts exist (multiple attempts to update the same packet action). Finally, the packet Action-Profiles are resolved at the end of the packet processing pipe by updating packet attributes and generating copies.

BROADCOM®