KeyStone Architecture II Packet Accelerator 2 (PA2) for K2E and K2L Devices

User's Guide

TEXAS INSTRUMENTS

Literature Number: SPRUHZ2
August 2014

Contents

List of Figures

List of Tables

Preface

About This Manual

The Packet Accelerator 2 (PA2) is one of the main components of the Network Coprocessor (NetCP) for K2E and K2L Devices peripheral. PA2 works together with the Security Accelerator 2 (SA2) and the Gigabit Ethernet (GbE) Switch Subsystem for K2E and K2L Devices to form a network processing solution. The purpose of PA2 in the NetCP is to perform packet processing operations such as packet header classification, checksum generation, and multi-queue routing.

Notational Conventions

This document uses the following conventions:

- Commands and keywords are in **boldface** font.
- Arguments for which you supply values are in *italic* font.
- Terminal sessions and information the system displays are in screen font.
- Information you must enter is in **boldface screen font**.
- Elements in square brackets ([ ]) are optional.

Notes use the following conventions:

NOTE: Means reader take note. Notes contain helpful suggestions or references to material not covered in the publication.

The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

CAUTION

Indicates the possibility of service interruption if precautions are not taken.

WARNING

Indicates the possibility of damage to equipment if precautions are not taken.

Related Documentation from Texas Instruments

Gigabit Ethernet (GbE) Switch Subsystem for K2E and K2L Devices User's Guide	SPRUHZ3
Interrupt Controller (INTC) for KeyStone Devices User'sGuide	SPRUGW4
Multicore Navigator for KeyStone Devices User's Guide	SPRUGR9
Network Coprocessor (NetCP) for K2E and K2L Devices User's Guide	SPRUHZ0
Security Accelerator 2 (SA2) for K2E and K2L Devices User's Guide	SPRUHZ1

Chapter 1
SPRUHZ2 – August 2014

Introduction

This chapter describes high-level features and block diagram of the Packet Accelerator 2 architecture.

Topic **Page**

1.1 Purpose of the Peripheral

The Packet Accelerator 2 (PA2) is one of the main components of the Network Coprocessor for K2E and K2L devices (NetCP) peripheral. PA2 works together with the Security Accelerator 2 (SA2) and the Gigabit Ethernet Switch Subsystem for K2E and K2L Devices to form a network processing solution. The PA2 sub-system is enhanced to support fully-offloaded, fast-path operations in both ingress and egress directions to provide the following functionality:

- Ethernet and SRIO packet classification
- Stateless L3/L4 Firewall (ACL)
- Outer and inner IP packet classification
- Outer and inner IP reassembly
- TCP/UDP/GTP-U based LUT2 classification
- IPv4 and TCP/UDP checksum generation and verification
- SCTP or custom CRC generation and verification
- Programmable system statistics
- Post-Classification operation such as packet patch, protocol header and/or trailer removal
- Egress or forwarding traffic flow cache operations
- Inner IP L3/L4 patching
- Inner IP fragmentation
- Outer IP insertion/update
- Pre-IPSEC and Post-IPSEC processing
- Outer IP fragmentation
- L2 header insertion/update

1.2 Features

The PA2 hardware features listed below are utilized in the PA LLD firmware to provide functional modules as shown in Section 1.3:

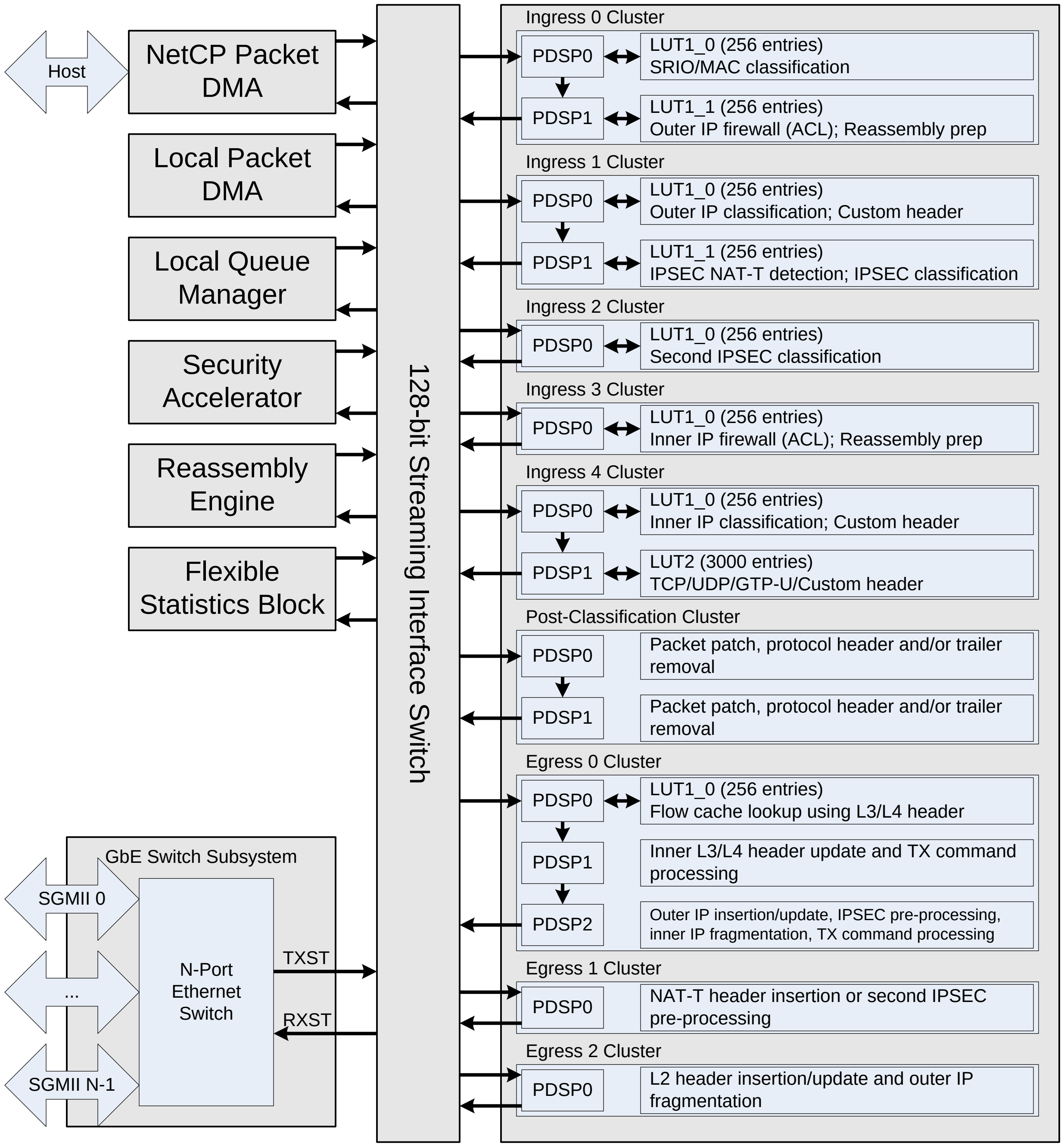
- High performance packet DMA controller
  - 21 transmit channels
  - 91 receive channels
- Packet accelerator for packet lookup and modification operations
  - Contains 15 independent packed data structure processors (PDSP)
  - Provides eight 1st pass lookup table (LUT1) accelerators
    - Each instance supports up to 256 entries
    - All lookup operations occur at 1.5 million packets per second
    - Lookup result based on field match scoring (based on RFC4301)
  - One 2nd pass lookup table accelerator
    - Supports up to 3,000 entries
    - All lookup operations occur at 1.5 million packets per second
    - Lookup can occur while add or delete operation is ongoing
  - Fifteen Chunk Data Engines (CDE)
    - Operates as tightly coupled data modification accelerators to the PDSPs
    - Includes Checksum / CRC acceleration
      - Capable of generating and checking IP- / UDP-level checksums
      - Capable of generating and checking CRCs
      - Supports programmable polynomial up to 32 bit
  - Fifteen 16-bit timers for use by PDSPs/CDEs

1.3 Functional Block Diagram

Figure 1-1 shows the PA2 functional block diagram for its intended use cases. PA2 hardware components are utilized in the PA LLD firmware to provide the following functional modules:

- **Ingress 0 Cluster**
  - CDE 0, PDSP0, and 256-entry LUT1_0: SRIO/MAC header parsing and classification
  - CDE 1, PDSP1, and 256-entry LUT1_1: L3/L4 header parsing and pre-IPSEC firewall (ACL) lookup
- **Ingress 1 Cluster**
  - CDE 0, PDSP0, and 256-entry LUT1_0: Outer IP or custom header parsing and classification
  - CDE 1, PDSP1, and 256-entry LUT1_1: IPSEC NAT-T detection, IPSEC header parsing and classification
- **Ingress 2 Cluster**
  - CDE 0, PDSP0, and 256-entry LUT1_0: 2nd IPSEC Header parsing and classification
- **Ingress 3 Cluster**
  - CDE 0, PDSP0, and 256-entry LUT1_0: L3/L4 header parsing and post-IPSEC firewall (ACL) lookup
- **Ingress 4 Cluster**
  - CDE 0, PDSP0, and 256-entry LUT1_0: Inner IP or custom header parsing and classification
  - CDE 1, PDSP1, and 3000-entry LUT2: TCP/UDP/GTP-U/Custom header parsing and LUT2 lookup
- **Post-Classification Cluster**
  - CDE 0 and PDSP0: Post-classification processing
  - CDE 1 and PDSP1: Post-classification processing
- **Egress 0 Cluster**
  - CDE 0, PDSP0, and 256-entry LUT1_0: Inner L3/L4 header parsing and Flow cache lookup
  - CDE 1 and PDSP1: Inner L3/L4 Header update and Tx command processing
  - CDE 2 and PDSP2: Outer IP insertion/update, IPSEC pre-processing, inner IP fragmentation and Tx command processing
- **Egress 1 Cluster**
  - CDE 0 and PDSP0: NAT-T header insertion or second IPSEC pre-processing
- **Egress 2 Cluster**
  - CDE 0 and PDSP1: L2 header insertion/update and outer IP fragmentation

Figure 1-1. Packet Accelerator 2 Functional Block Diagram


Host
NetCP Packet DMA
Local Packet DMA
Local Queue Manager
Security Accelerator
Reassembly Engine
Flexible Statistics Block
GbE Switch Subsystem
SGMII 0
...
SGMII N-1
N-Port Ethernet Switch
TXST
RXST
128-bit Streaming Interface Switch
Ingress 0 Cluster
PDSP0
LUT1_0 (256 entries) SRIO/MAC classification
PDSP1
LUT1_1 (256 entries) Outer IP firewall (ACL); Reassembly prep
Ingress 1 Cluster
PDSP0
LUT1_0 (256 entries) Outer IP classification; Custom header
PDSP1
LUT1_1 (256 entries) IPSEC NAT-T detection; IPSEC classification
Ingress 2 Cluster
PDSP0
LUT1_0 (256 entries) Second IPSEC classification
Ingress 3 Cluster
PDSP0
LUT1_0 (256 entries) Inner IP firewall (ACL); Reassembly prep
Ingress 4 Cluster
PDSP0
LUT1_0 (256 entries) Inner IP classification; Custom header
PDSP1
LUT2 (3000 entries) TCP/UDP/GTP-U/Custom header
Post-Classification Cluster
PDSP0
Packet patch, protocol header and/or trailer removal
PDSP1
Packet patch, protocol header and/or trailer removal
Egress 0 Cluster
PDSP0
LUT1_0 (256 entries) Flow cache lookup using L3/L4 header
PDSP1
Inner L3/L4 header update and TX command processing
PDSP2
Outer IP insertion/update, IPSEC pre-processing, inner IP fragmentation, TX command processing
Egress 1 Cluster
PDSP0
NAT-T header insertion or second IPSEC pre-processing
Egress 2 Cluster
PDSP0
L2 header insertion/update and outer IP fragmentation

Architecture

This chapter describes the details of the Packet Accelerator architecture.

Topic **Page**

2.1 Clock Control

The Packet Accelerator 2 (PA2) has one clock, which it receives from the Network Coprocessor (NetCP) for K2E and K2L Devices. This clock is used to operate all of the PA2 logic, plus the common NetCP logic, such as the packet DMA controller and the packet streaming switch; therefore, the PA2 clock domain must also be enabled when using the Security Accelerator 2 (SA2) or the Gigabit Ethernet (GbE) Switch Subsystem for K2E and K2L Devices. To reduce power consumption, this clock is disabled by default; therefore, this clock must be enabled before using the PA2. For more information about this clock, including the operating frequency, see the *Network Coprocessor (NetCP) for K2E/K2L Devices User's Guide* (SPRUGZ6), and the device-specific data manual. For more information about power management for the PA2, please see Section 2.8.

2.2 Packet Accelerator 2 Transmit Queue Interface

When sending packets to PA2 from the Host, packets must be sent through the Multicore Navigator queue interface. There are 11 hardware TX queues that are used to send packets to PA2 for processing, with each queue directly mapped to a specific cluster, the Reassembly engine (RA), or the Flexible Statistics Block. When a descriptor (with linked packet) is pushed into one of the TX queues, the NetCP packet DMA automatically transfers the packet to the specific engine in PA2. The mapping between the queues and the PA2 hardware processing engines is shown in the device specific user guide. For more information about the packet DMA and queues, see the multicore navigator user guide.

2.3 Programming the Packet Accelerator 2 with Low-Level Driver

To ease the task of programming the PA2 many of the hardware details are abstracted by a low level driver (LLD) software package that is provided in the Multicore Software Development Kit (MCSDK). Included with the PA LLD are firmware images that must be loaded onto the PDSPs in each cluster before using PA2. Due to interdependencies between the firmware and the PA LLD, all users must use the PA LLD. Failure to use the PA LLD will result in undefined behavior.

NOTE: Firmware images loaded into the PA2 PDSPs are provided by TI in the MCSDK and no custom firmware will be supported.

To minimize the software migration effort of existing applications from older devices, the new PA LLD for the PA2 maintains backward compatibility of all existing APIs with a few minor exceptions. Even though the PA LLD APIs are backwards compatible with previous devices some minor application changes may also be necessary due to transport layer changes: more NetCP PKTDMA channels and queues, the addition of a local PKTDMA and queue manager in the PA2, and the global NetCP PKTDMA queue layout. Also, some features that are supported for backward compatibility should be deprecated or changed due to the more advanced capability of the PA2. The PA2 provides several new features such as outer IP and inner IP reassembly, egress flow and flow cache operation, pre and post IPSEC stateless firewall (ACL) operation, as well as a local PKTDMA responsible for moving packets between PA2 components which are all supported through the PA LLD with new API calls.

For PA LLD documentation that discusses the aforementioned topics please download the MCSDK package for your device and refer to the mcsdk_03_xx_xx_xx\pdk_keystone2_x_xx_xx_xx\packages\ti\drv\pa\docs folder. The release notes in the docs folder contain a release entry that discusses the new APIs and changes for the PA2 described above. The doxygen files in the docs folder will serve as the documentation for the PA LLD APIs.

For provided code examples that use the PA LLD see the mcsdk_03_xx_xx_xx\pdk_keystone2_x_xx_xx_xx\packages\exampleProjects folder and look for folder names prefixed with PA that also contain you device specific name in them, such as K2E or K2L.

2.4 Packet Accelerator Components

The PA2 contains different hardware modules to provide efficient Ethernet packet manipulation. The sub-sections below give brief descriptions of these modules. The PA LLD abstracts the implementation of these modules so the information below is provided to give clarity when the PA LLD mentions a hardware component. For a visualization of the components refer to the PA2 block diagram in Figure 1-1.

2.4.1 Cluster Components

The ingress, post processing, and egress clusters are where most of the packet classification and modification occurs in the PA2. The following subsections are a description of each of the components that make up the clusters. Once again, these descriptions are only provided for clarity in the case that the PA LLD mentions a cluster component. An intimate knowledge of this architecture is not necessary to interact with the PA2 at the abstraction level provided by the PA LLD. For a visualization of the components in each cluster refer to the diagrams in Section 2.5.

2.4.1.1 Splitter/Combiner Module (SPLITCOMB)

From a packet classification and modification perspective only the first and last portions of a packet are considered interesting. The start of packet (SOP) contains the header information and end of packet (EOP) contains the CRC information. The middle of the packet (MOP) contains the packet payload data that is not necessary for packet classification and most modification procedures.

The primary job of the SPLITCOMB module is to split the incoming packet into SOP, MOP, and EOP upon entry to the cluster and then at cluster exit to combine the three parts back together again into a full packet. When split, the SOP and EOP data gets sent to the next processing block in the cluster while the MOP data gets sent to a shared packet buffer. This allows the processing blocks in the cluster to have high speed access to the SOP and EOP data but also allows access to the MOP data in the shared buffer if needed. When combined, the SOP and EOP data is received at the SPLITCOMB by the final processing block in the cluster. This SOP and EOP data is then combined with the MOP data from the shared buffer to recreate the original, or newly modified, packet.

In addition to the Ethernet packet data described in the previous paragraph, there is a small amount of sideband packet information attached to each packet that is used for routing and modification information. The SPLITCOMB also separates this data and provides it to the next processing block during cluster ingress and then combines this data back with the Ethernet packet data on cluster egress.

2.4.1.2 Chunk Data Engine (CDE)

The most common and time consuming operations performed during packet classification and modification are reading, writing, and manipulating data within a chunk buffer. For this reason, the chunk data engine (CDE) was created to offload much of this work from the PDSPs within the PA2 clusters.

The CDE receives the SOP, EOP, and sideband packet info from the previous component in the cluster and presents a sliding window view of the data to the associated PDSP. The PDSP controls the movement of the sliding window through the CDE data. Once the sliding window is in place, the contents of the window can be read into the PDSP or written by the PDSP in a single cycle.

2.4.1.3 Packed Data Structure Processor (PDSP)

The PDSP is a processor which is optimized for performing embedded tasks that require manipulation of packed memory mapped data structures such as networking packet headers, descriptors, and hardware control / status registers.

The PDSP views the SOP, EOP, and packet info sideband data through the sliding window view presented by its associated CDE. The PDSP then performs a query on its associated lookup table (LUT) if it has one. If an LUT entry matches the current packet then the routing and modification information from that LUT entry is written into the packet info sideband data. If the PDSP does not have an associated LUT then it is used for packet modifications determined by the values read from the packet info sideband data: header updates/insertions/removals/, fragmentation, etc. Check the diagrams for each cluster in Section 2.5 to see each PDSPs function.

2.4.1.4 Lookup Tables (LUT)

There are nine lookup tables (LUTs) used by the PA2 for packet classification. The nine tables are split between eight LUT1 lookup tables each with up to 256 entries and one LUT2 lookup table with up to 3,000 entries. Table entries contain matching criteria as well as packet routing or modification information. If a packet is identified to match all of the criteria of the entry then the routing and/or modification information of the entry is added to the packet info sideband data of the packet. The routing information is used to determine the next destination for the packet: the host, discard, the next PA2 cluster, etc. The modification information is used to determine if another component in the PA2 should perform its modification on the packet: CRC generation, header insertion/removal, etc. For the specific functions of each of the nine LUTs see the cluster diagrams in Section 2.5.

2.4.1.4.1 LUT1

The LUT1 tables are used for SRIO, Ethernet L2/L3/L4, and custom packet classifications. LUT1 entries contain multiple fields that can either be matched or considered as a wildcard. For example, you could add an Ethernet Layer 2 LUT1 entry that matches on a specific source MAC address and a specific VLAN ID but set all of the other fields of the layer 2 header as a wildcard. So, as long as a packet's source MAC address and VLAN ID matched the values in your table entry, then that packet would be a match regardless of any of the other values in the layer 2 header. The routing and modification information from the matching entry would then be applied to the packet.

LUT1 table entries are added through PA LLD function calls. The Pa_addSrio, Pa_addMac2, Pa_addIp2, Pa_addAcl, and Pa_addFc functions all create LUT1 entries and insert them into the LUT1 tables.

NOTE: It is important to note that when using the LUT1 that entries are searched in order when looking for a match. Since match fields can be wildcarded, multiple table entries could be a match for a single packet. For this reason it is required that software ensure that entries in the LUT1 are made in order from the most general matching criteria to the most specific.

Some example LUT1 matching fields follow. For the full list see the PA LLD Doxygen in the docs folder mentioned in Section 2.3.

- SRIO
  - Source ID
  - Destination ID
  - Priority
- Ethernet Layer 2
  - Source MAC address
  - Destination MAC address
  - VLAN ID
  - EtherType
  - MPLS tag
- Ethernet Layer 3
  - Source IP address
  - Destination IP address
  - IP Protocol (IPv4/IPv6)
  - DSCP value
- Ethernet Layer 4
  - Source Port Range (for ACL firewall)
  - Destination Port Range (for ACL firewall)

2.4.1.4.2 LUT2

The LUT2 table is used for Ethernet L4 and custom packet classifications. For Ethernet layer 4 matching, the PA LLD supports exact matching on destination port number which can be used for protocols such as TCP, UDP, and GTP-U. Similar to LUT1, once a table match occurs, the routing and modification information in the table entry will be added to the matched packet's packet info sideband data. See the PA LLD Doxygen in the docs folder mentioned in Section 2.3 for more information.

LUT2 table entries are added through PA LLD function calls. The Pa_addPort2 and Pa_addCustomLUT2 functions create LUT2 entries and insert them into the LUT2 table. Since LUT2 table entries do not contain wildcard fields each packet may only match exactly one table entry. Due to this fact, the order that LUT2 table entries are added does not matter. Software may add the entries in any order.

2.4.1.5 Packet Checker Module (CHECKER)

Depending on the cluster, or the function that was performed by a CDE/PDSP, certain types of CRC may need to be performed on portions of the packet data. The Checker module is responsible for these CRC calculations. The Checker receives the SOP, EOP and packet info sideband data from the CDE. The received packet info contains flags that dictate if the Checker needs to calculate a CRC and if so, what type of CRC. The Checker component is capable of accessing the MOP data in the shared packet buffer if the data is needed to calculate the requested CRC. Even though the Checker module calculates the CRC it does not modify the any of the packet data. The Checker module simply places the CRC result in the packet info sideband data and forwards it, along with the SOP and EOP, on to the next CDE in the cluster. If the packet info flags did not request a CRC to be calculated then the Checker module just passes all the data on to the next CDE. See the PA LLD for more information on configuring the Checker modules in each cluster.

2.4.2 Reassembly Engine (RA) Module

The Reassembly Engine (RA) is capable of reassembling fragmented IP packets without the need for host intervention in PA2. If a fragmented packet is detected in ingress cluster 0 (outer IP fragmentation) or ingress cluster 3 (inner IP fragmentation) then that packet will be forwarded to the RA for packet reassembly. Once a fragmented packet is detected in a data flow the remainder of the packets in that data flow will also be forwarded to the RA. This is to ensure that the packet order of the data flow will be maintained inside of the PA2. Once the RA has reassembled the packets they will be passed to the next ingress cluster for further processing (either ingress 1 or ingress 4). For more information on the RA please see the *Network Coprocessor (NetCP) for K2E/K2L Devices User's Guide*.

2.4.3 Flexible Statistics Block (FSB) Engine

The Flexible Statistics Block Engine (FSB) in the PA2 keeps track of several different statistics within different PA2 components and can be queried using the PA LLD to return the results to the host. Below is a small subset of the statistics that are available to users, for the full list see the PA LLD doxygen documentation referenced in Section 2.3.

- LUT1 Classification Statistics
  - Number of packets entering LUT1 classification PDSPs
  - Number of IPv4 packets
  - Number of IPv6 packets
  - Number of packets with a table match
  - Number of packets without a table match
  - Number of ingress fragmented IP packets
  - Number of packets discarded
- LUT2 Classification Statistics
  - Number of packets entering the LUT2 classification PDSP
  - Number of UDP packets
  - Number of TCP packets
  - Number of packets discarded

- Egress Cluster
  - Number of packets matching an egress flow cache
- ACL Statistics
  - Number of packets that matched ACL rules
  - Total bytes in the matched ACL packets
- Reassembly Engine Statistics
  - Number of successfully reassemble packets
  - Number of IP fragments with zero-byte payload
  - Number of IP fragments which are discarded due to incorrect fragment length

2.5 Cluster Architecture

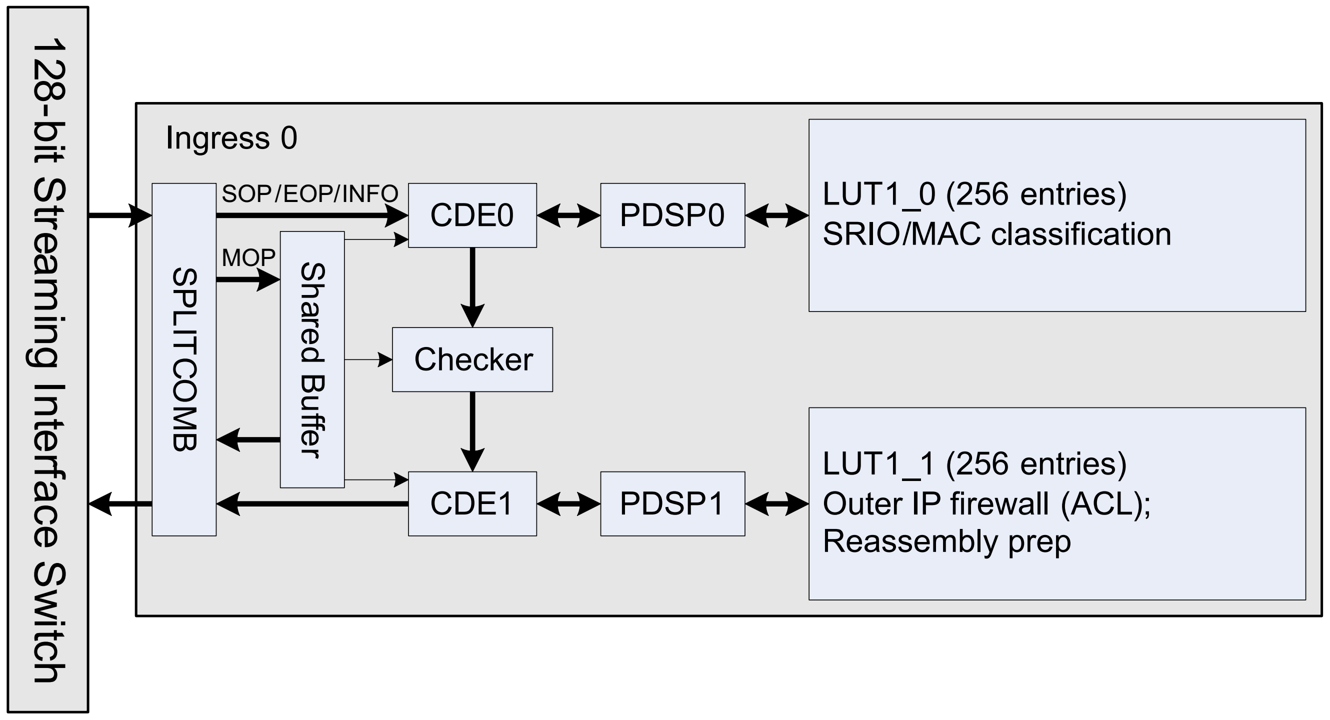
The PA2 SPLITCOMB, CDEs, PDSPs, LUTs, and CHECKER modules along with the associated firmware provide functional units called Clusters consisting of five ingress stages (Ingress0-4), a post-processing stage (Post-Classification) and three egress stages (Egress 0-2). Each stage has its intended function, which is described briefly in the sub-sections below. Ingress packets (from the Ethernet Switch through PA2 to the host) are expected to follow the flow Ingress 0 → Ingress 1 → Ingress 2 → Ingress 3 → Ingress 4 → Post → Host. Egress packets (from the host through PA out the switch) are expected to follow the flow Egress 0 → Egress 1 → Egress 2 → Ethernet Switch. Ingress packets can be directly routed to egress path without host intervention. The packets can also be routed between PA2 and the Security Accelerator (SA) sub-systems multiple times to perform encryption, decryption and authentication operation. It is also possible for packets to be routed to the host at any point during the ingress process if the application does not need to take advantage of all of the functions that the ingress path has to offer. Packets may also be sent directly from the host to the Ethernet switch, bypassing the PA2 egress process, if so desired. The data flows available in the PA2 are extremely flexible and are determined by the programming of the PA2 with the PA LLD.

2.5.1 Cluster 0: Ingress 0 Architecture

Ingress 0 is intended for L2 parsing and L3/L4 Firewall (ACL) operations. Figure 2-1 shows a block diagram of the cluster. It performs the following functions:

- L2 Header Parsing
- Custom Parsing
- L3/L4 Header Parsing
- Outer IP Firewall (ACL)
- Reassembly Engine Preparation

Figure 2-1. Cluster 0: Ingress 0 Functional Block Diagram

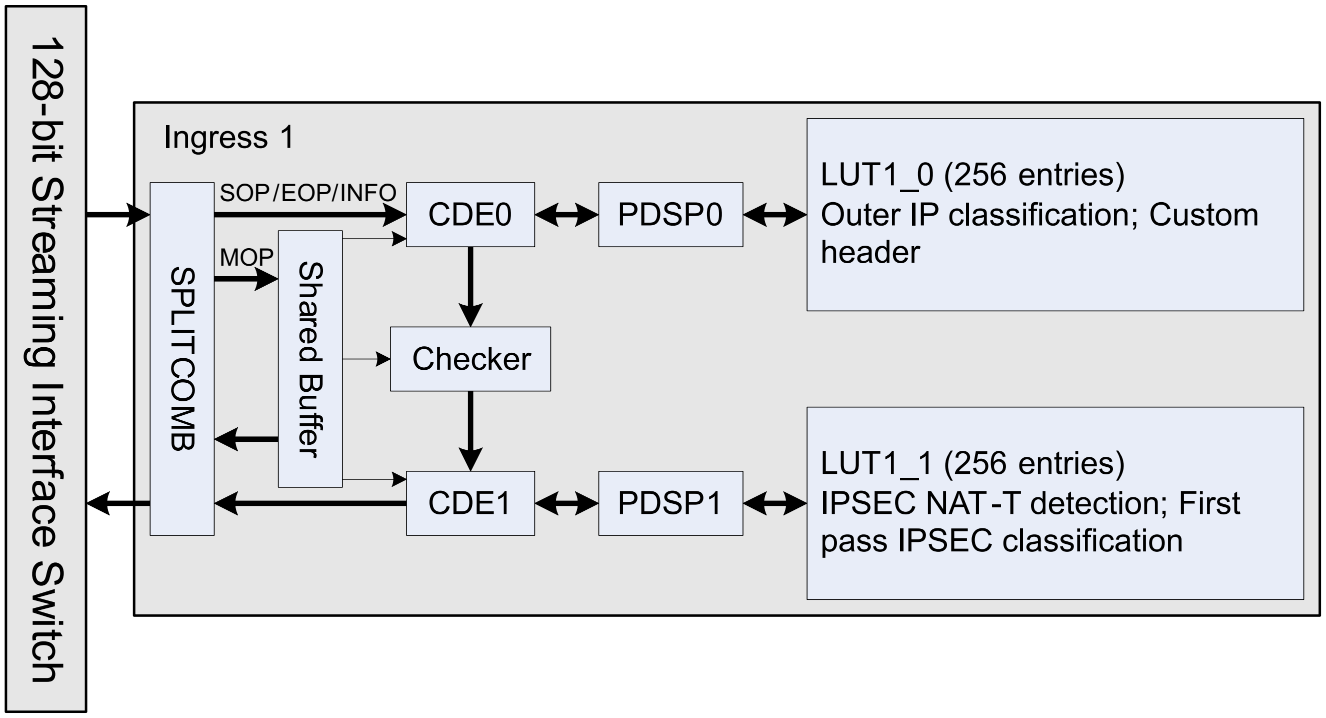


2.5.2 Cluster 1: Ingress 1 Architecture

Ingress 1 is intended for L3/L4 Classification and IPSEC preparation. Figure 2-2 shows a block diagram of the cluster. It performs the following functions:

- L3/L4 Classification
- NAT-T detection
- IPSEC Pass 1 Preparation

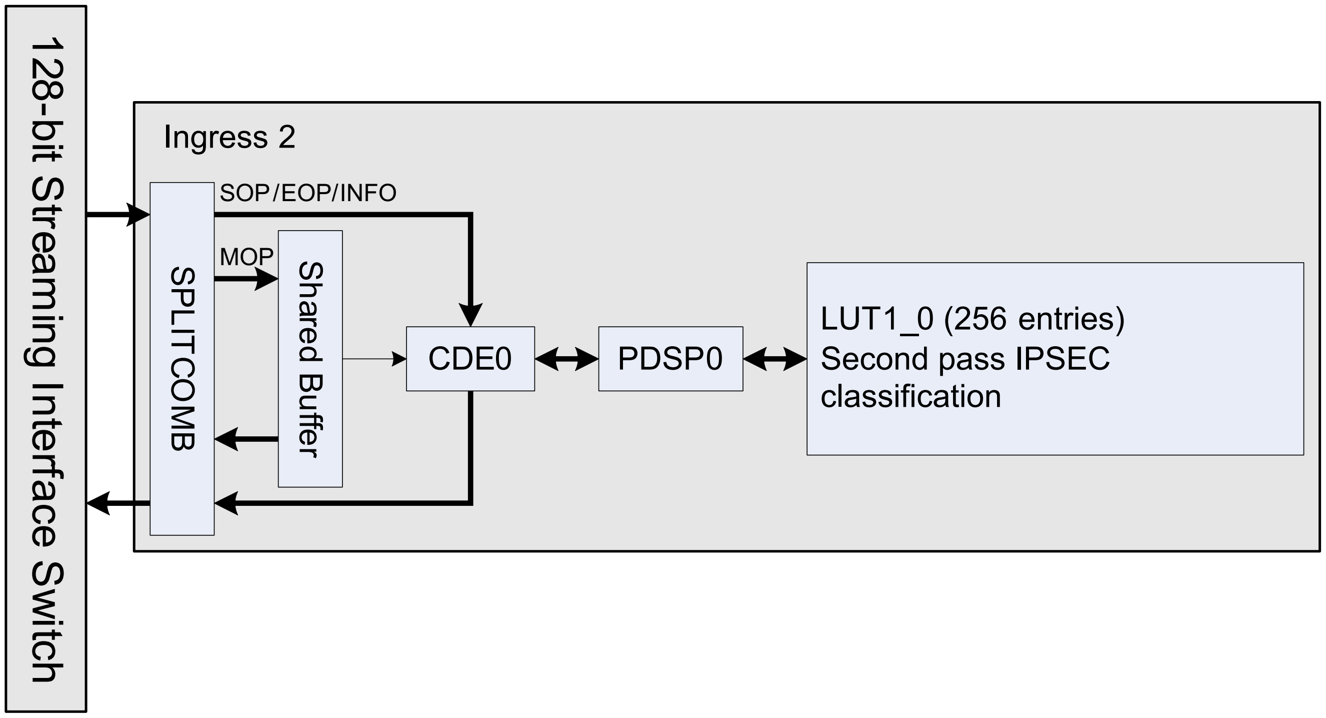
Figure 2-2. Cluster 1: Ingress 1 Functional Block Diagram


128-bit Streaming Interface Switch
Ingress 1
SOP/EOP/INFO
MOP
SPLITCOMB
Shared Buffer
CDE0
PDSP0
Checker
CDE1
PDSP1
LUT1_0 (256 entries)
Outer IP classification; Custom header
LUT1_1 (256 entries)
IPSEC NAT-T detection; First pass IPSEC classification

2.5.3 Cluster 2: Ingress 2 Architecture

Ingress 2 is intended for IPSEC Pass 2 preparation. Figure 2-3 shows a block diagram of the cluster.

Figure 2-3. Cluster 2: Ingress 2 Functional Block Diagram

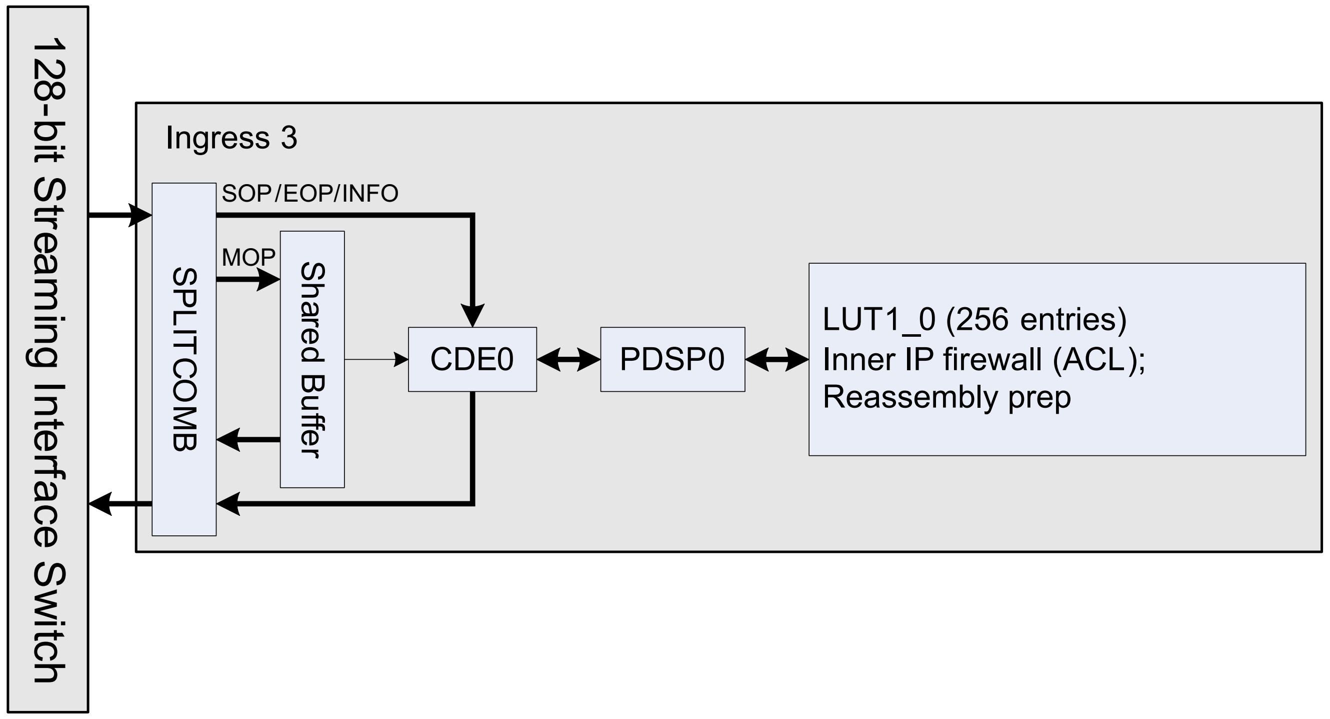


2.5.4 Cluster 3: Ingress 3 Architecture

Ingress 3 is intended for SA Cleanup and Inner IP Firewall (ACL). Figure 2-4 shows a block diagram of the cluster. It performs the following functions:

- Security Accelerator Cleanup
- L3/L4 Header Parsing
- Inner IP Firewall (ACL)
- Reassembly Engine Preparation

Figure 2-4. Cluster 3: Ingress 3 Functional Block Diagram

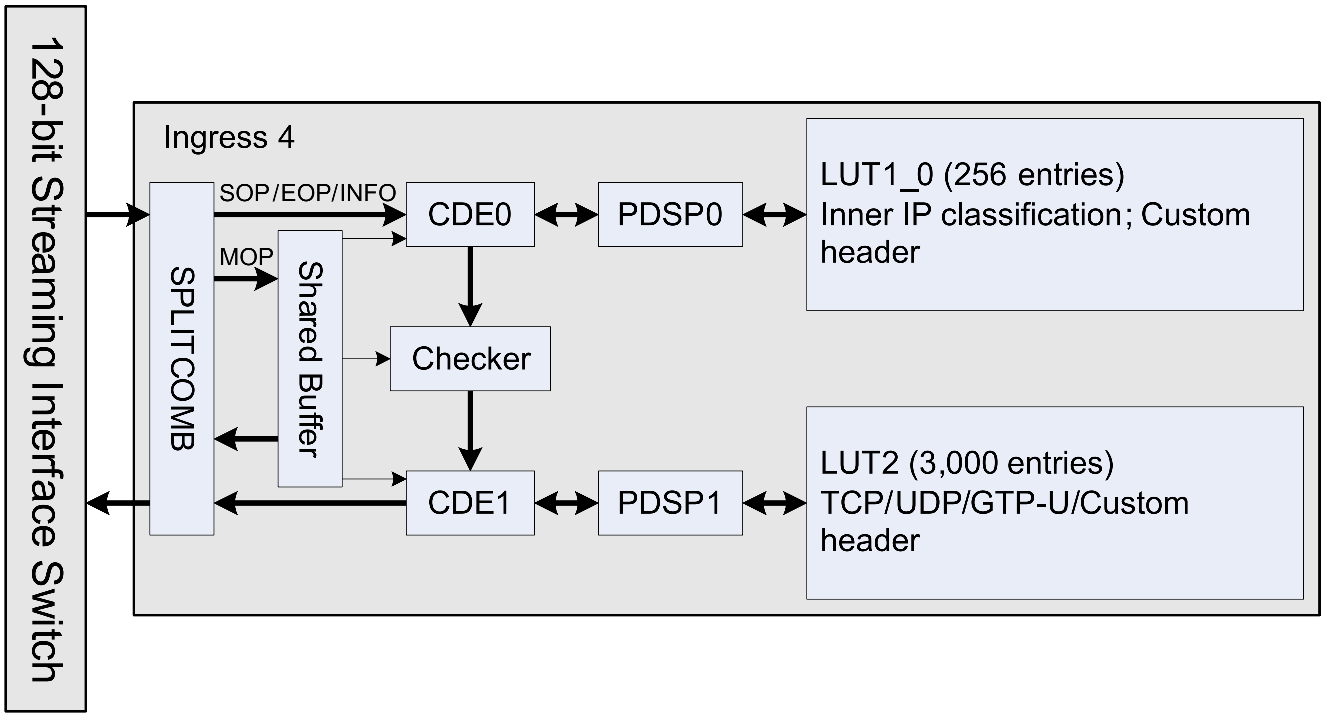


2.5.5 Cluster 4: Ingress 4 Architecture

Ingress 4 is intended for packet checking and air cipher preparation. It also contains a CRC engine. Figure 2-5 shows a block diagram of the cluster. It performs the following functions:

- L3/L4 Final Parsing
- L4 Checksum
- CRC
- SCTP Preparation
- L4 Processing
- Security Accelerator Air Cipher Preparation

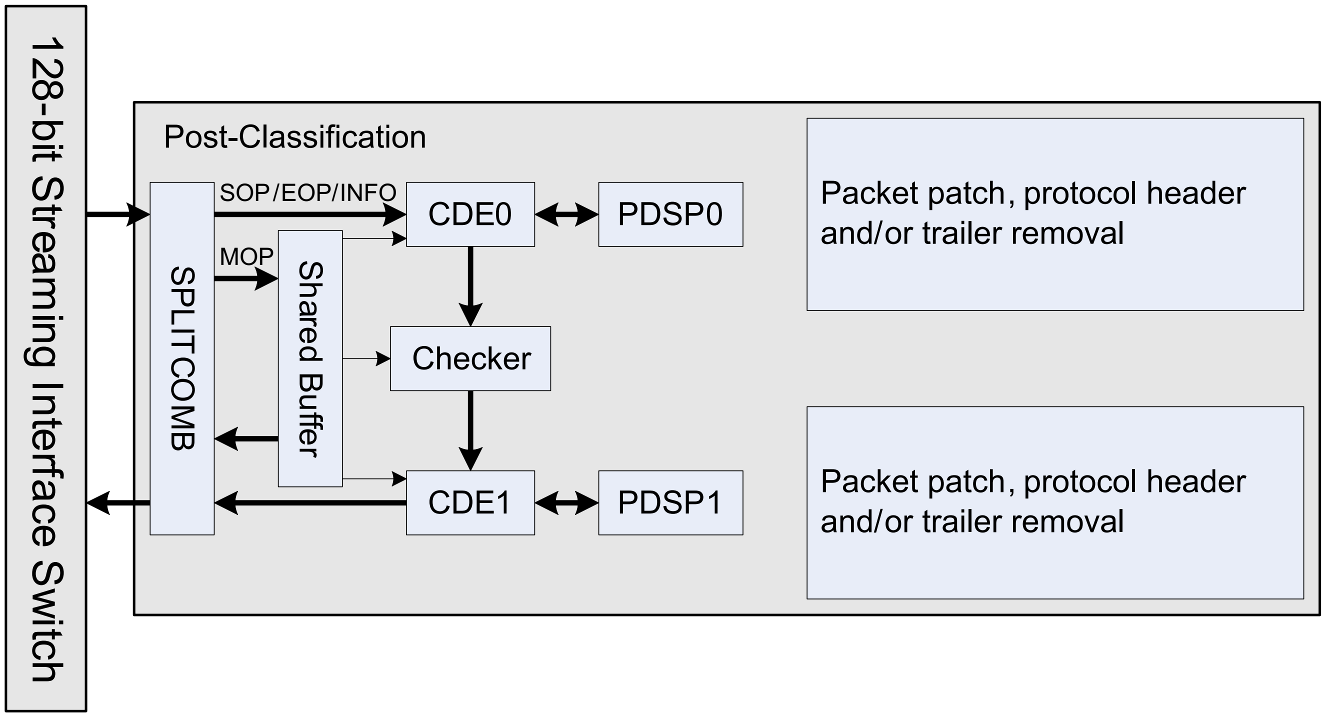
Figure 2-5. Cluster 4: Ingress 4 Functional Block Diagram



2.5.6 Cluster 5: Post-Classification Architecture

Figure 2-6 shows a block diagram of the post-classification cluster.

Figure 2-6. Cluster 5: Post-Classification Functional Block Diagram

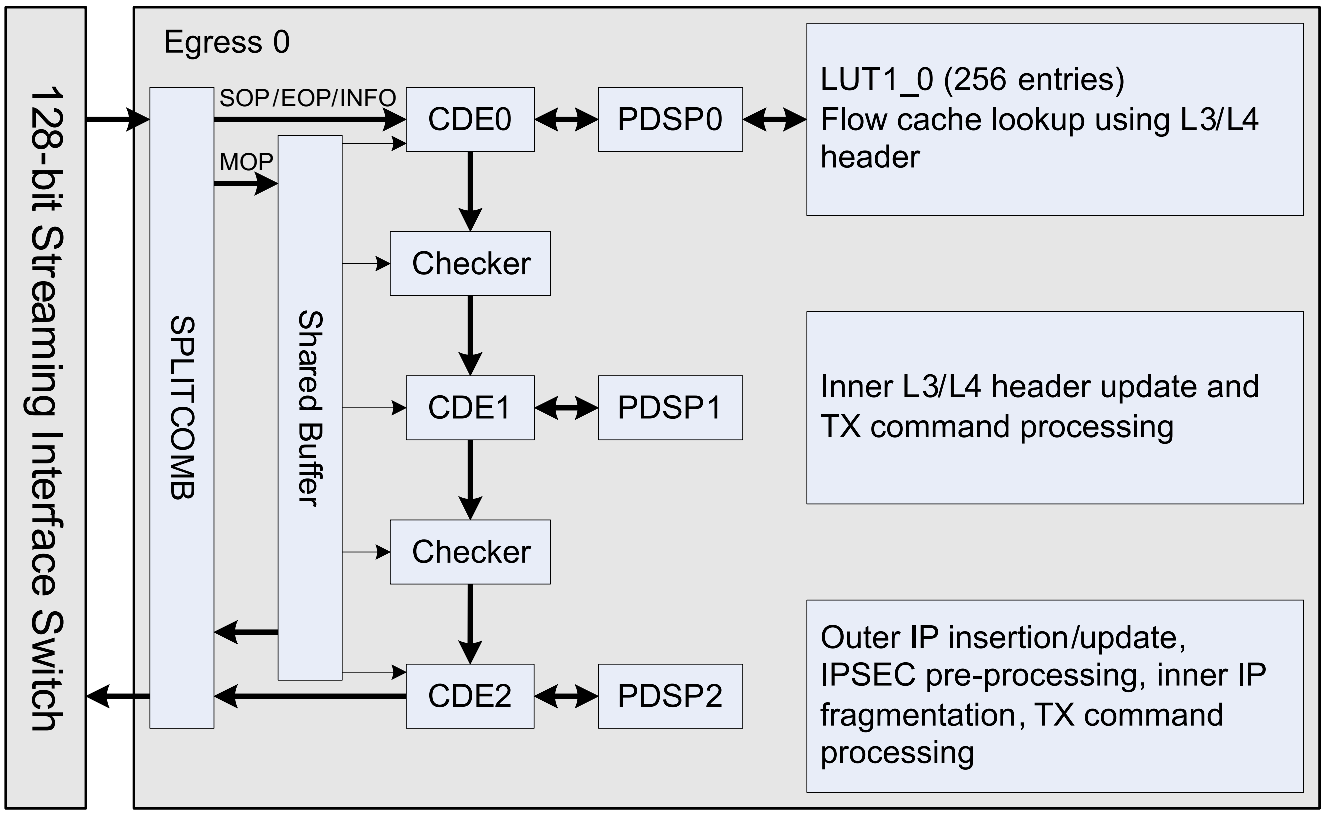

128-bit Streaming Interface Switch
Post-Classification
SPLITCOMB
SOP/EOP/INFO
MOP
Shared Buffer
CDE0
PDSP0
Checker
CDE1
PDSP1
Packet patch, protocol header and/or trailer removal
Packet patch, protocol header and/or trailer removal

2.5.7 Cluster 6: Egress 0 Architecture

Egress 0 is intended for Flow Cache, egress preparation and fragmentation. Figure 2-7 shows a block diagram of the cluster. It performs the following functions:

- Flow Cache
- L4 Checksum
- CRC Preparation
- IP Fragmentation
- IPSEC Pass1 (Inner IP) Preparation

Figure 2-7. Cluster 6: Egress 0 Functional Block Diagram

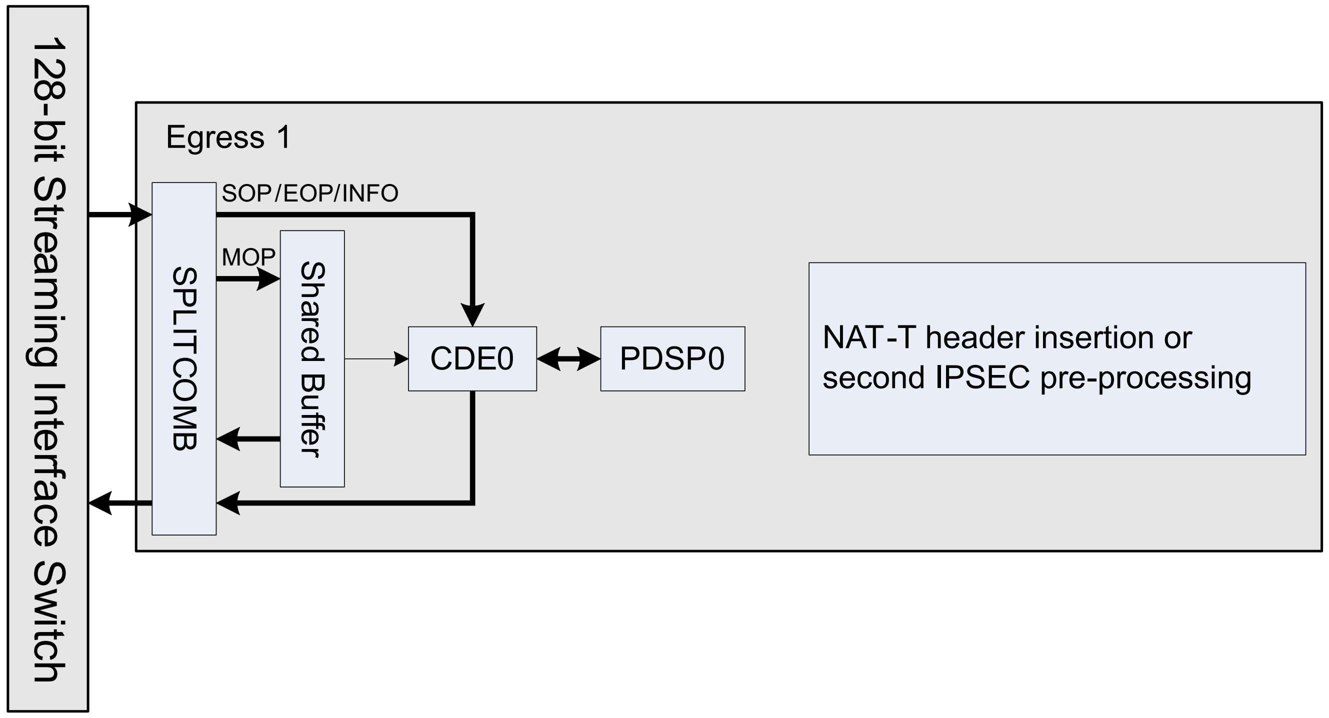

128-bit Streaming Interface Switch
Egress 0
SOP/EOP/INFO
MOP
SPLITCOMB
Shared Buffer
CDE0
PDSP0
LUT1_0 (256 entries)
Flow cache lookup using L3/L4 header
Checker
CDE1
PDSP1
Inner L3/L4 header update and TX command processing
Checker
CDE2
PDSP2
Outer IP insertion/update, IPSEC pre-processing, inner IP fragmentation, TX command processing

2.5.8 Cluster 7: Egress 1 Architecture

Egress 1 is intended for Outer IP IPSEC preparation. Figure 2-8 shows a block diagram of the cluster. It performs the following functions:

- Security Accelerator Patching
- IPSEC Pass 2 (External IP) Preparation

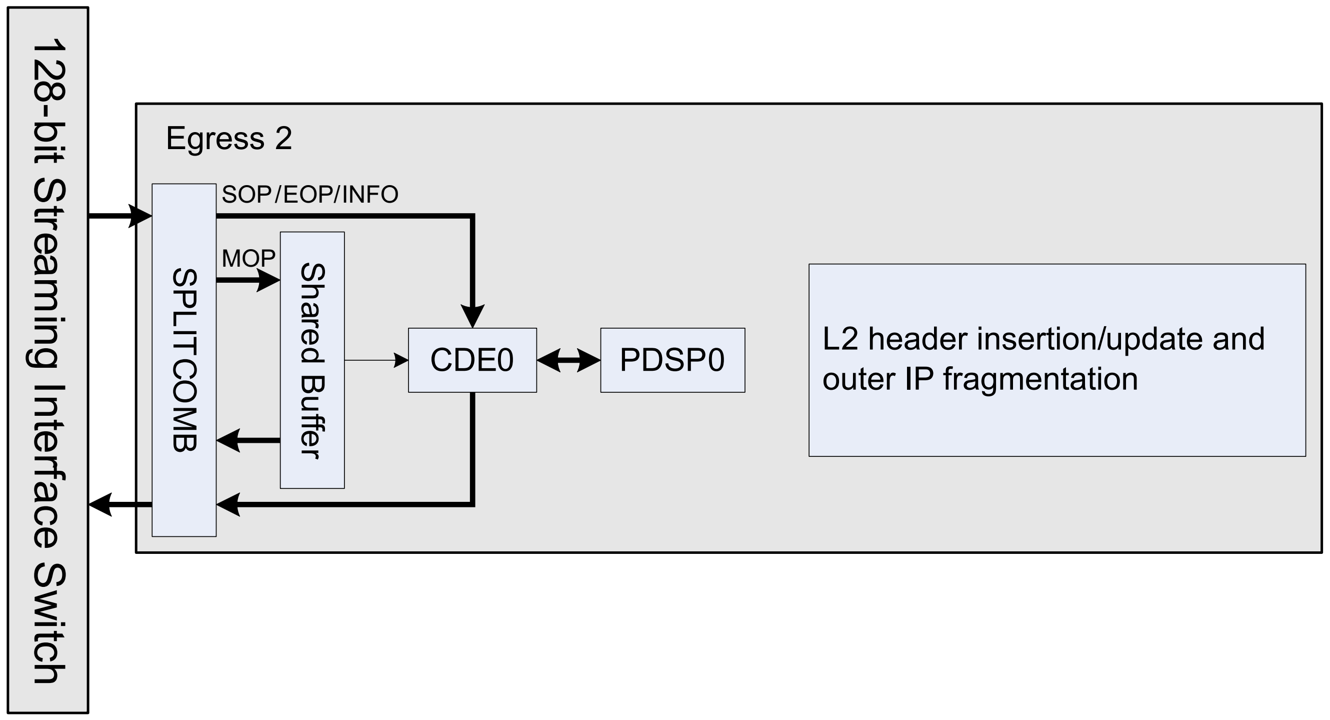
Figure 2-8. Cluster 7: Egress 1 Functional Block Diagram


128-bit Streaming Interface Switch
Egress 1
SOP/EOP/INFO
MOP
SPLITCOMB
Shared Buffer
CDE0
PDSP0
NAT-T header insertion or second IPSEC pre-processing

2.5.9 Cluster 8: Egress 2 Architecture

Egress 2 is intended for final IP Fragmentation and L2 Framing. Figure 2-9 shows a block diagram of the cluster.

Figure 2-9. Cluster 8: Egress 2 Functional Block Diagram


128-bit Streaming Interface Switch
Egress 2
SOP/EOP/INFO
MOP
SPLITCOMB
Shared Buffer
CDE0
PDSP0
L2 header insertion/update and outer IP fragmentation

2.6 Adding Entries to the Classification Engines

This section will discuss adding entries to the classification engines. Before the classification engines can be used to classify packets, entries must be added to the classification engines' lookup tables. Entries are configurable, and are registered by sending a configuration packet to the desired configuration engine. All configuration packets must be generated using the PA LLD, which contains several APIs for adding entries (see the PA LLD documentation for more information). The PA2 also provides the ability to link entries between classification engines. This allows the ability to specify a specific combination of L2/L3/L4 addresses. For example, a UDP address in the L4 classify engine can be linked to an IP address in the L3 classify engine, which can then be linked to a MAC address in the L2 classify engine. The PA LLD provides many code examples for adding entries to the lookup tables, but a high level overview is provided in Section 2.6.1.

2.6.1 Adding Entries to the Classification Engines

Step 1. Configure the header fields that the classification engine should use to classify the packet

Step 2. Configure the routing information that the classification engine will use to route the packet after classification

Step 3. Optionally, configure the link to an entry in a preceding classification engine

Step 4. Use the PA LLD to generate a configuration packet (e.g. Pa_addIp2)

Step 5. Link the configuration packet to a descriptor to prepare the packet to be sent to the PA2

Step 6. Set the protocol specific information words to communicate to the classification engine that this is a configuration packet

Step 7. Optionally set the descriptor software information words

Step 8. Push the configuration packet onto the transmit queue for the desired classification engine (this information is provided by the PA LLD when the configuration packet is generated)

Step 9. The PA classification engine will send a configuration response packet to indicate whether or not the entry was successfully added to the classification engine. The user can wait for the response to proceed, or the user can go on with other tasks until the configuration response arrives.

Step 10. Check that the entry was successfully added to the classification engine and register it with the PA LLD by using the Pa_forwardResult API to allow the PA LLD to examine the response packet.

2.7 DMA Event Support

All DMA events for PA2 are handled through the packet streaming switch and the packet DMA controller in the NetCP. For more information about the packet streaming switch, see the *Network Coprocessor (NetCP) for K2E/K2L Devices User's Guide* (SPRUGZ6). For more information about the and the packet DMA controller, see the *Network Coprocessor (NetCP) for K2E/K2L Devices User's Guide* (SPRUGZ6) or the *Multicore Navigator for KeyStone Devices User's Guide* (SPRUGR9).

2.8 Power Management

PA2 power is managed through the NetCP power domain and through disabling the clock that operates the PA2 logic. PA2 shares its power domain with the other modules in the NetCP and therefore, can not be powered down independently of NetCP. The clock for PA2 is disabled by default, and can be controlled independently of the other modules in the NetCP. For more information about power management for the PA2, see the *Network Coprocessor (NetCP) for K2E/K2L Devices User's Guide* (SPRUGZ6) or the device-specific data manual.

Data Flow Examples

This chapter provides examples of the data flow within the Packet Accelerator 2 (PA2), and interaction with other modules in the Network Coprocessor (NetCP) for K2E and K2L Devices.

Topic **Page**

3.1 Overview

The following sections cover examples of the flow of data packets through the hardware in the packet accelerator 2 (PA2) and Network Coprocessor (NetCP) for K2E and K2L Devices. Since the PA2 is a highly configurable, the data flow is dependent on how the PA2 has been setup. This section covers the following examples:

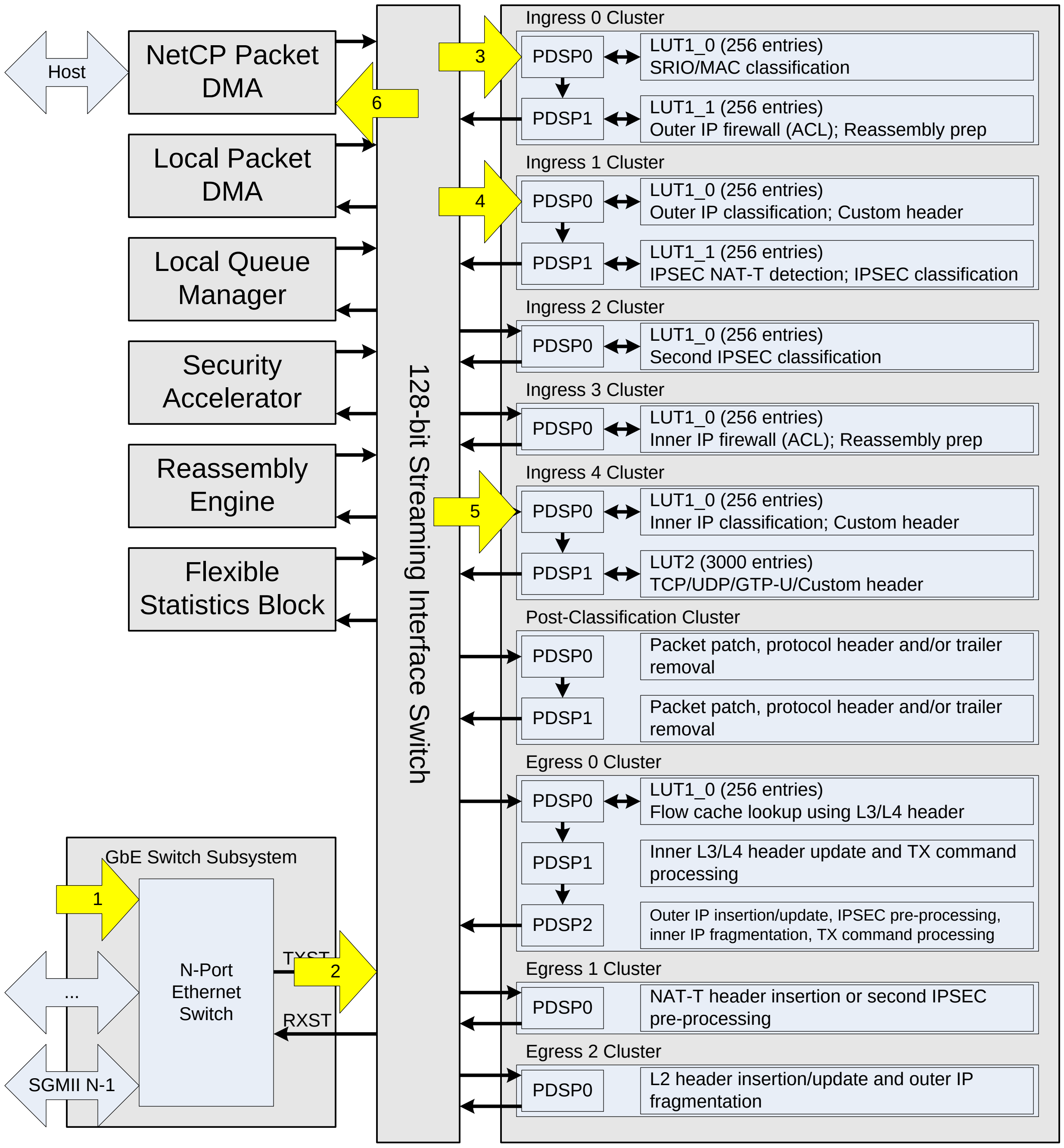
- Receive Classification Example Section 3.2
- Transmit Checksum Generation Example Section 3.3

Note that PA2 may have additional processing features that require different PA LLD functions not described in the data flow examples below. These examples are meant to introduce the overall configuration procedure and interaction of PA2 with other NetCP sub-systems. For more information regarding additional configurations, refer to the PA LLD doxygen documentation.

3.2 Receive Classification Example

The following section covers an example of receive packet routing in the PA2 for a non-encrypted data packet. The packet originates from the gigabit Ethernet (GbE) switch subsystem, and will show one example of routing the packet through the PA2 to the host.

Figure 3-1. Receive Classification Example


Host
NetCP Packet DMA
Local Packet DMA
Local Queue Manager
Security Accelerator
Reassembly Engine
Flexible Statistics Block
GbE Switch Subsystem
N-Port Ethernet Switch
TXST
RXST
...
SGMII N-1
1
2
3
4
5
6
128-bit Streaming Interface Switch
Ingress 0 Cluster
PDSP0
LUT1_0 (256 entries) SRIO/MAC classification
PDSP1
LUT1_1 (256 entries) Outer IP firewall (ACL); Reassembly prep
Ingress 1 Cluster
PDSP0
LUT1_0 (256 entries) Outer IP classification; Custom header
PDSP1
LUT1_1 (256 entries) IPSEC NAT-T detection; IPSEC classification
Ingress 2 Cluster
PDSP0
LUT1_0 (256 entries) Second IPSEC classification
Ingress 3 Cluster
PDSP0
LUT1_0 (256 entries) Inner IP firewall (ACL); Reassembly prep
Ingress 4 Cluster
PDSP0
LUT1_0 (256 entries) Inner IP classification; Custom header
PDSP1
LUT2 (3000 entries) TCP/UDP/GTP-U/Custom header
Post-Classification Cluster
PDSP0
Packet patch, protocol header and/or trailer removal
PDSP1
Packet patch, protocol header and/or trailer removal
Egress 0 Cluster
PDSP0
LUT1_0 (256 entries) Flow cache lookup using L3/L4 header
PDSP1
Inner L3/L4 header update and TX command processing
PDSP2
Outer IP insertion/update, IPSEC pre-processing, inner IP fragmentation, TX command processing
Egress 1 Cluster
PDSP0
NAT-T header insertion or second IPSEC pre-processing
Egress 2 Cluster
PDSP0
L2 header insertion/update and outer IP fragmentation

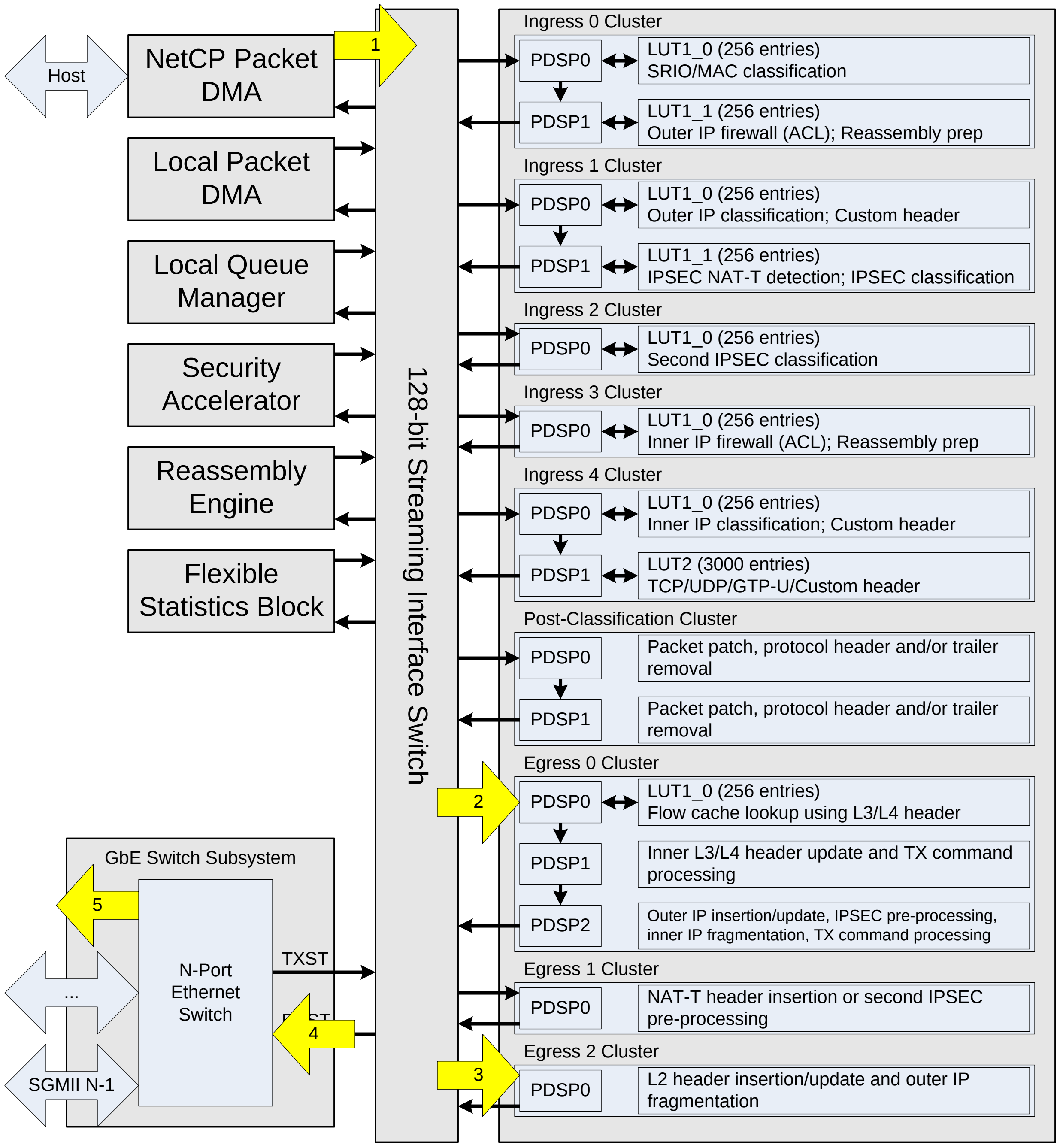
This example assumes that Ingress 0, Ingress 1, and Ingress 4 lookup tables are setup to match the fields for each header, and that the packets that arrive have MAC, IPv4, and UDP headers. The Pa_addMac2, Pa_addIp2, and Pa_addPort2 functions from the PA LLD would be used to add the LUT entries in this example data flow.

1. A packet is received at the GbE Switch Subsystem on Ethernet port 1. The packet is forwarded onto host port 0 in the GbE switch.
2. The packet is routed out of host port 0 of the GbE switch towards the streaming interface switch. Based on the configuration of the streaming interface, packets from host port 0 will be routed to the ingress 0 cluster.
3. The packet arrives at the ingress 0 cluster for first stage classification. The packet MAC address and EtherType match an entry in the LUT1_0 lookup table. The routing information from the matched table entry sends the packet for further classification. Since LUT1_1 for ACL is not configured in this example the traffic will pass through the second PDSP in the cluster.
4. The packet arrives at the ingress 1 cluster for next stage classification. The IPv4 address in the packet match a table entry in the LUT1_0 lookup table. The routing information from the matched table entry sets the destination of the packet for further processing at the LUT2 table. The traffic will bypass ingress 1 PDSP1 because the destination is now set to ingress 4 for LUT2 classification.
5. The packet arrives at the ingress 4 cluster for further classification. Since the next processing header is set to UDP in the previous step, the packet will pass through the first PDSP in the cluster. The UDP port number from the packet matches a table entry in the LUT2 table. The routing information from the matched table entry sets the destination of the packet to the host. The flow number and queue number are also set by the matched table entry.
6. The packet arrives at the NetCP PKTDMA where the flow number and queue number determine which free queue the descriptor will come from as well as which receive queue the descriptor will be placed on.

3.3 Transmit Checksum Generation Example

The following section covers an example of transmit packet routing in the PA2 for a non-encrypted data packet. This shows one example of routing transmit packets through the PA2 to the gigabit Ethernet (GbE) switch subsystem. The packet originates from the host, and will be sent to the egress clusters for a UDP checksum before being sent out over the network.

Figure 3-2. Transmit Checksum Generation Example


Host
NetCP Packet DMA
Local Packet DMA
Local Queue Manager
Security Accelerator
Reassembly Engine
Flexible Statistics Block
1
128-bit Streaming Interface Switch
Ingress 0 Cluster
PDSP0
LUT1_0 (256 entries) SRIO/MAC classification
PDSP1
LUT1_1 (256 entries) Outer IP firewall (ACL); Reassembly prep
Ingress 1 Cluster
PDSP0
LUT1_0 (256 entries) Outer IP classification; Custom header
PDSP1
LUT1_1 (256 entries) IPSEC NAT-T detection; IPSEC classification
Ingress 2 Cluster
PDSP0
LUT1_0 (256 entries) Second IPSEC classification
Ingress 3 Cluster
PDSP0
LUT1_0 (256 entries) Inner IP firewall (ACL); Reassembly prep
Ingress 4 Cluster
PDSP0
LUT1_0 (256 entries) Inner IP classification; Custom header
PDSP1
LUT2 (3000 entries) TCP/UDP/GTP-U/Custom header
Post-Classification Cluster
PDSP0
Packet patch, protocol header and/or trailer removal
PDSP1
Packet patch, protocol header and/or trailer removal
Egress 0 Cluster
2
PDSP0
LUT1_0 (256 entries) Flow cache lookup using L3/L4 header
PDSP1
Inner L3/L4 header update and TX command processing
PDSP2
Outer IP insertion/update, IPSEC pre-processing, inner IP fragmentation, TX command processing
Egress 1 Cluster
PDSP0
NAT-T header insertion or second IPSEC pre-processing
Egress 2 Cluster
3
PDSP0
L2 header insertion/update and outer IP fragmentation
GbE Switch Subsystem
5
N-Port Ethernet Switch
TXST
4
...
SGMII N-1

This example assumes that the packet has MAC, IPv4, and UDP headers. The IPv4 and UDP header values are assumed to match a flow cache entry in the LUT1_0 table in the egress 0 cluster. The example also assumes that the flow cache entry that matches the packet calls for a UDP checksum calculation and that no other insertions or modifications are requested by the entry.

1. Host processor generates a packet and pushes it to the transmit queue that corresponds to the egress 0 cluster. The NetCP Packet DMA takes the packet from the transmit queue and sends it to the egress 0 cluster through the Streaming Interface Switch. To find the specific queue number that corresponds to the egress 0 cluster see your device specific user guide.
2. The packet arrives at the egress 0 cluster for flow cache classification. The packet IP address and UDP port match a flow cache entry in the LUT1_0 lookup table. The flow cache entry directs the packet to have a UDP checksum generated by the egress process. The packet info of the packet is updated with these instructions and the packet is forwarded on to PDSP1 in the cluster. The UDP checksum command is issued by PDSP1 and the calculated checksum is placed at the UDP header. PDSP2 will perform outer IP insertion and pre-IPSEC operations if required. None of the modifications from the egress 1 cluster are needed, so the packet is sent out to the Streaming Interface with the new UDP checksum toward the final egress cluster for potential out IP fragmentation.
3. The packet arrives at the egress 2 cluster and the configured MTU value is checked to see if outer IP fragmentation is necessary. If the packet is larger than the MTU value, the packet will be fragmented. The packet then passes through to the Streaming Switch Interface on its way to the GbE Switch Subsystem.
4. The packet arrives at host port 0 of the GbE Switch.
5. The packet is forward to Ethernet port 1 in the switch and sent out over the Ethernet with the new UDP checksum.

NOTE: Calculating and inserting a UDP checksum is just one of a vast number of modifications or updates that the egress clusters can perform on a packet. If fully configured, the flow cache entry can specify up to four levels of packet modifications. To learn more about the capabilities of the egress clusters along with flow cache see the PA LLD doxygen documentation referenced in Section 2.3.

Chapter 4
SPRUHZ2–August 2014

Registers

This chapter describes the registers available in the Packet Accelerator 2 (PA2) and its submodules. For clarity, the registers for each module and submodule are described separately. Provided for each register is a bit field description and a memory offset address. The offset address values provided are relative to the associated base address of the PA module.

See the *Network Coprocessor (NetCP) for K2E and K2L Devices User's Guide* (SPRUGZ6) for the base address of the Packet Accelerator (PA2) module relative to the NetCP.

Table 4-1 lists the Packet Accelerator 2 registers.

Topic **Page**

Table 4-1. Packet Accelerator 2 Memory Map

Address Offset [(1)]	Size	Region	Section
000000h	256	PDSP mailbox slots	Section 4.1
006000h	256	Flexible Statistics Block Registers	Section 4.3
008000h	32K	Flexible Statistics Block Memory	Section 4.4

(1) The actual addresses of these registers are device specific. See your device-specific data manual to verify the PA register addresses.

4.1 PDSP Mailbox Slots

This section describes the PDSP Mailbox Slots available in the Packet Accelerator (PA2). The register address offsets listed in Table 4-2 are relative to the PDSP Mailbox Slots memory region. To determine the base address of PDSP Mailbox Slots region relative to the PA2 memory map, see Table 4-1.

The PDSP mailbox slots are for use by system masters in the PA2 to pass information to the PDSPs. Non-zero writes to the mailbox slots will cause the PDSP to be notified on the input status pin the cycle following the write. Any data stored in the mailbox slots can be read by the PDSPs.

Table 4-2. PDSP Mailbox Slots

Address Offset	Status Bit Routing
00h	PDSP 0 Mailbox Registers
10h	PDSP 1 Mailbox Registers
20h	PDSP 2 Mailbox Registers
30h	PDSP 3 Mailbox Registers
40h	PDSP 4 Mailbox Registers
50h	PDSP 5 Mailbox Registers
60h	PDSP 6 Mailbox Registers
70h	PDSP 7 Mailbox Registers
80h	PDSP 8 Mailbox Registers
90h	PDSP 9 Mailbox Registers
A0h	PDSP10 Mailbox Registers
B0h	PDSP 11 Mailbox Registers
C0h	PDSP 12 Mailbox Registers
D0h	PDSP 13 Mailbox Registers
E0h	PDSP 14 Mailbox Registers
F0h	Reserved

4.2 Reassembly Engine

The reassembly engine details are provided in the *Network Coprocessor (NetCP) for K2E/K2L Devices User's Guide* (SPRUGZ6).

4.3 Flexible Statistics Block Registers

This section describes the registers in the Flexible Statistics Block (FSB) available in PA2. The register address offsets listed in Table 4-3 are relative to the FSB register region. To determine the base address of the FSB memory region relative to the PA2 memory map, please see Table 4-1.

Table 4-3. Flexible Statistics Block Registers Memory Map

Address Offset	Register	Section
00h	FSB Revision Register	Section 4.3.1
04h	FSB Soft Reset Register	Section 4.3.2
08h	Enable and Allocation Control Register	Section 4.3.3
0Ch	Counter Update Register	Section 4.3.4
10h	Control Timer Register	Section 4.3.5
14h	Load Timer Register	Section 4.3.6
18h	Value Timer Register	Section 4.3.7
1Ch	Statistics Packet Routing Register	Section 4.3.8

4.3.1 FSB Revision Register (FSB_REV)

The FSB Revision Register contains the major and minor revisions for the module.

Table 4-4. FSB Revision Register (FSB_REV)

Bits	Field	Type	Reset	Description
31-30	FSB_SCHEME	R	1h	Scheme
29-28	Reserved	R	0h	Reserved
27-16	FSB_MODID	R	EF5h	Module ID
15-11	FSB_REVRTL	R	0h	RTL revision. Will vary depending on release.
10-8	FSB_REVMAJ	R	1h	Major revision
7-6	FSB_REVCUST	R	0h	Custom revision
5-0	FSB_REVMIN	R	0h	Minor revision

4.3.2 FSB Soft Reset Register (FSB_SOFT_RESET)

The FSB Soft Reset Register is written in order to clear the contents of the FSB.

Table 4-5. FSB Soft Reset Register (FSB_SOFT_RESET)

Bits	Field	Type	Reset	Description
31-0	TRIGGER	W	0h	Writing anything to this field causes the command FIFOs to be emptied, all statistics to be cleared and all bit masks to be reset to 1.

4.3.3 Enable and Allocation Control Register (EA_CONTROL)

The EA_CONTROL Register contains the enable for the engine and controls the allocation of 64-bit counters in the memory. It is possible to configure the engine to only have 32-bit counters by setting the 64BITCNT field to zero.

Table 4-6. Enable and Allocation Control Register (EA_CONTROL)

Bits	Field	Type	Reset	Description
31	ENABLE	RW	0h	Enables stat engine. If the module is not in enabled state, no stats increment is processed and both input and output streaming interfaces are disabled.
30-14	Reserved	R	0h	Reserved.
13-0	64BITCNT	RW	200h	Defines the number of 64-bit counters in the memory (must be even number). If this number multiplied by 8-byte is less than 16KB (stats memory space), then the remaining space is allocated for 32-bit counters.

4.3.4 Counter Update Register (COUNT_UPDATE)

The COUNT_UPDATE Register defines the events that may reset the counter value. The ‘GENPKT’ bit provides a way to manually trigger the generation of output packet. Writing ‘1’ to both GENPKT and GLBCLR bit in this register has the same effect as sending a command packet with bit map of all zeroes and the SW0 ‘Send Update’ bit set. Writes to update bit 0 and bit 31 while the stats engine is enabled may corrupt counter values.

Table 4-7. Counter Update Register (COUNT_UPDATE)

Bits	Field	Type	Reset	Description
31	CLR_OUTPKT	RW	0h	Clear after every out packet bit: Define the counter reset mode after a stats packet is generated. This field has no effect during a manual (MMR) read. 0 = Keep running count and counter clear is controlled by bit mask 1 = reset all counters to zeroes.
30-2	Reserved	RW	0h	Reserved.
1	GENPKT	RW	0h	Generate packet bit: Writing ‘1’ to this bit will generate a packet of the latest statistics (regardless of status of timer). The gen packet bit remains set until the output phase is complete.
0	GLBCLR	RW	0h	Global clear bit: Writing ‘1’ to this bit will clear the currently active bit masks to 0h, which will effectively clear all counters on the next update. GLBCLR remains set until the output phase of the next update is complete. If ‘GENPKT’ bit is set as well, a statistics packet is generated. If ‘GENPKT’ bit is not set, the effect of GLBCLR does not take place until the next update.

4.3.5 Control Timer Register (CNTL_TIM)

The Control Timer Register controls the 16-bit count down timer module.

Table 4-8. Control Timer Register (CNT_TIM)

Bits	Field	Type	Reset	Description
31-16	Reserved	R	0h	Reserved.
15	PRESCALE_EN	RW	0h	Prescaler Enable. • 0 = Prescaler Disabled • 1 = Prescaler Enabled
14-6	Reserved	R	0h	Reserved.
5-2	PRESCALE	RW	Fh	Timer Prescale Value. These bits are a power-of-two prescaler values. The timer divides the frequency of the timer reference clock by a factor of 2 to 8192. 0000 = Divide by 2 0001 = Divide by 4 0010 = Divide by 8 0011 = Divide by 16 0100 = Divide by 32 0101 = Divide by 64 0110 = Divide by 128 0111 = Divide by 256 1000 = Divide by 512 1001 = Divide by 1024 1010 = Divide by 2048 1011 = Divide by 4096 1100 = Divide by 8192 1101-1111 = Disable timer
1	MODE	RW	0h	Timer Mode. • 0 = Sets the timer in one shot mode. After the counter expires, the START bit is set to 0, stopping the counter. • 1 = Sets the timer in auto-load mode. After the counter expires, the counter is reloaded with the value in the LOAD_TIM register and the timer resumes counting.
0	START	RW	0h	Timer Start. • 0 = Stops the 16-bit timer • 1 = Starts the 16-bit timer

4.3.6 Load Timer Register (LOAD_TIM)

The Load Timer Register contains the 16-bit count down value for timer module.

Table 4-9. Load Timer Register (LOAD_TIM)

Bits	Field	Type	Reset	Description
31-16	Reserved	R	0h	Reserved.
15-0	LOAD_TIM	RW	0h	Load Timer Value. This field is the 16-bit count down value for the timer module.

4.3.7 Value Timer Register (VALUE_TIM)

The Value Timer Register contains the current 16-bit count value of the timer.

Table 4-10. Value Timer Register (VALUE_TIM)

Bits	Field	Type	Reset	Description
31-16	Reserved	R	0h	Reserved.
15-0	VALUE	R	0h	Current Timer Value. This field contains the current 16-bit count value of the timer.

4.3.8 Statistics Packet Routing Info Register (STATPKT_INFO)

The Statistics Packet Routing Info Register needs to be set correctly so packet is sent to the destination where software expects it.

Table 4-11. Statistics Packet Routing Info Register (STATPKT_INFO)

Bits	Field	Type	Reset	Description
31	THREAD_ID	RW	0h	Go bit. Writing this bit will cause the add/delete operation to start. Once started, the operation cannot be cancelled. This bit will remain asserted until the add or delete operation completes.
30-26	PKT_TYPE	RW	0h	Define the packet type.
25-24	Reserved	R	0h	Reserved.
23-16	FLOW_INDEX	RW	0h	Define the flow index.
15-14	DST_QMGR	RW	0h	Define the destination queue manager.
13-12	Reserved	R	0h	Reserved.
11-0	DST_QUEUE	RW	0h	Define the queue number.

4.4 Flexible Statistics Block Memory

This section describes the statistics memory region in the Flexible Statistics Block (FSB) available in PA2. The memory address offsets listed in Table 4-12 are relative to the FSB memory region. To determine the base address of the FSB memory region relative to PA2 memory map, please see Table 4-1. Here N=0-4095.

Table 4-12. Flexible Statistics Block Memory Region

Address Offset	Register	Section
00h+N*4h	32-bit Statistics Value (Query Mode)	Section 4.4.1
4000h+N*4h	32-bit Statistics Value (Collect Mode)	Section 4.4.1

4.4.1 32-bit Statistics Value Memory (STAT_VAL_MEM)

The 32-bit Statistics Value Memory contains the current statistics values. When accessing the Query mode memory, the statistic values are not reset. If the Collect mode memory is accessed, the statistic values are reset to 0 after the read completes. To accommodate debug/verification, setting a counter value in STAT_VAL_MEM via the register write is also allowed. However if there is a statistic bump and a register write on the same statistic that occur simultaneously, the value in the register write may be ignored.

Table 4-13. 32-bit Statistics Value Memory (STAT_VAL_MEM)

Bits	Field	Type	Reset	Description
31-0	STATS_VALUE	RW	0h	32-bit statistic value at counter N.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		