



Mellanox NPS-400 EZdk

Release Notes

Software Ver. 18.0300.00

www.mellanox.com

NOTE:

THIS HARDWARE, SOFTWARE OR TEST SUITE PRODUCT (“PRODUCT(S)”) AND ITS RELATED DOCUMENTATION ARE PROVIDED BY MELLANOX TECHNOLOGIES “AS-IS” WITH ALL FAULTS OF ANY KIND AND SOLELY FOR THE PURPOSE OF AIDING THE CUSTOMER IN TESTING APPLICATIONS THAT USE THE PRODUCTS IN DESIGNATED SOLUTIONS. THE CUSTOMER’S MANUFACTURING TEST ENVIRONMENT HAS NOT MET THE STANDARDS SET BY MELLANOX TECHNOLOGIES TO FULLY QUALIFY THE PRODUCT(S) AND/OR THE SYSTEM USING IT. THEREFORE, MELLANOX TECHNOLOGIES CANNOT AND DOES NOT GUARANTEE OR WARRANT THAT THE PRODUCTS WILL OPERATE WITH THE HIGHEST QUALITY. ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT ARE DISCLAIMED. IN NO EVENT SHALL MELLANOX BE LIABLE TO CUSTOMER OR ANY THIRD PARTIES FOR ANY DIRECT, INDIRECT, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES OF ANY KIND (INCLUDING, BUT NOT LIMITED TO, PAYMENT FOR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY FROM THE USE OF THE PRODUCT(S) AND RELATED DOCUMENTATION EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



Mellanox Technologies
350 Oakmead Parkway Suite 100
Sunnyvale, CA 94085
U.S.A.
www.mellanox.com
Tel: (408) 970-3400
Fax: (408) 970-3403

© Copyright 2016. Mellanox Technologies Ltd. All Rights Reserved.

Mellanox®, Mellanox logo, Accelio®, BridgeX®, CloudX logo, CompustorX®, Connect-IB®, ConnectX®, CoolBox®, CORE-Direct®, EZchip®, EZchip logo, EZappliance®, EZdesign®, EZdriver®, EZsystem®, GPUDirect®, InfiniHost®, InfiniScale®, Kotura®, Kotura logo, Mellanox Federal Systems®, Mellanox Open Ethernet®, Mellanox ScalableHPC®, Mellanox TuneX®, Mellanox Connect Accelerate Outperform logo, Mellanox Virtual Modular Switch®, MetroDX®, MetroX®, MLNX-OS®, NP-1c®, NP-2®, NP-3®, Open Ethernet logo, PhyX®, PSIPHY®, SwitchX®, Tiler®, Tiler logo, TestX®, TuneX®, The Generation of Open Ethernet logo, UFM®, Virtual Protocol Interconnect®, Voltaire® and Voltaire logo are registered trademarks of Mellanox Technologies, Ltd.

All other trademarks are property of their respective owners.

For the most updated list of Mellanox trademarks, visit <http://www.mellanox.com/page/trademarks>

Table of Contents

1	Introduction	4
2	This Release	4
3	System Requirements	4
4	Documentation	5
5	Changes and New Features	5
5.1	General Changes and New Features	5
5.2	CPE Changes and New Features	5
5.3	DPE Changes and New Features	6
5.4	LDK Changes and New Features	6
5.5	Sample Application Changes and New Features	8
6	Bug Fixes	9
6.1	CPE Bug Fixes	9
6.2	LDK Bug Fixes	9
7	API Changes	10
7.1	API Changes in Rev 18.0300.00	10
7.1.1	CPE API Changes in Rev 18.0300.00	10
7.1.2	DPE API Changes in Rev 18.0300.00	13
8	Known Limitations	14
8.1	CPE Known Limitations	14
8.2	DPE Known Limitations	14
8.3	LDK Known Limitations	14

1 Introduction

The EZdk software development kit is a comprehensive set of design and runtime tools developing both data-plane and control-plane applications for NPS-400 devices.

This release provides post-silicon support for the NPS-400 network processor. It fully supports real chip bring-up and application development.

This document provides additional information related to the release not contained in the product's manuals. It is recommended that you read this document before using the product. These release notes outline the modifications employed since EZdk release 2.1a Open.

2 This Release

This release of the EZdk software development kit for NPS-400 devices is aimed at allowing customers to develop NPS data-plane and control plane applications. This release includes the following main components:

- EZldk – Linux® development kit enabling customers to develop and build the Linux kernel, file system and GNU tool-chain for the NPS device.
- EZdp – Data-plane service library providing the APIs used to develop data-plane applications on the NPS device.
- EZcp – Control-plane service library providing the APIs used to develop control-plane applications for the NPS device.
- Sample applications demonstrating how to develop applications for the NPS-400 devices using EZdk.

This release represents work in progress, and is subject to change in future releases.

3 System Requirements

EZdk-nps is compliant with RedHat6.x distribution or any derivative such as CentOS6.x. Any other environment, such other versions of the distribution or different distributions are not guaranteed to work.



Note: The GCC toolchain is provided with EZdk as precompiled binaries. Since it is dynamically linked to standard libraries of distribution and built upon RedHat6.x using the toolchain, it may not work on other environments due to missing libraries.

4 Documentation

The following documentation can be found in the doc folder of the EZdk installation:

General

- *EZdk Release Notes* (this document) – provides additional information related to the release that is not contained in the product’s manuals.
- *EZdk Installation Guide* – provides information on EZdk installation procedure and contents.

Data Plane Environment and Control Plane Environment

- *NPS-400 Developer’s Guide* – describes each of the control plane libraries and each of the data plane libraries and their respective APIs. Describes the various EZdk Control Plane Library (EZcp) components and how they can be ported to various platforms and used to develop control-plane applications for NPS-based products.

Sample Applications

- *NPS-400 Demo Application* – sample data plane and control plane application designed to enable developers to quickly begin writing an application for an NPS-400 system.

5 Changes and New Features

Following are the main new features and functionality introduced in this release (since EZdk release 2.1a Open).

5.1 General Changes and New Features

The EZdk product number was changed for this release. In the previous release it was 2.1a Open. This release it is 18.0300.00, where 18 is the EZdk OpenNPU product number (i.e. fixed number) and the other digits represent the software release number that changes with each release.

5.2 CPE Changes and New Features

Table 1: CPE Changes and New Features

No.	Description
EZcp	
1.	EZapiChannel_PCIInfo was removed.
2.	Number of logical IDs available to the user was reduced from 256 to 255. Logical ID 255 is used by the DP application (EZframe).
3.	Removed PCI advance NIC capabilities support.
4.	Removed error queue support.
5.	Allow PMU sequence numbering in PMU queue basis; removed per RX channel sequence numbering.
6.	Allow loopback interfaces rate of up to 100Gbps.
7.	Force buffer allocation on IMEM when RX allocation profile mode is IMEM only.

No.	Description
8.	System Info is enabled by default.
9.	Optimized the load partition command by enhancing the reset of search related NPS memory.
10.	Optimized search multiple NPS operations by supporting channel interleaving in memory operations.
11.	Increased the number of supported search structures from 128 to 256.
12.	Removed the possibility for a recycle race condition on DP updates of dynamic hash. This includes not allowing sharing of a hash designated index pool for more than one structure. Also using the same pool for both signature and entry page is no longer allowed.
13.	Added support for direct table with an 8 byte result in NPS internal memory.
EZc2p	
14.	New Config to Python (EZc2p) library in the /cpe/c2p/ folder allows users to export their system control plane configuration into a Python file.
EZspy	
15.	Monitor simple for monitoring single defined field at memory or register.

5.3 DPE Changes and New Features

Table 2: DPE Changes and New Features

No.	Description
EZdp	
1.	Aging pool APIs were removed.
2.	PCI Express interface APIs were removed.

5.4 LDK Changes and New Features

Table 3: LDK Changes and New Features

No.	Description
1.	The 5 pct auxiliary registers of the core can be accessed through an ioctl call through the dpl module. The options for the ioctl calls of the read and write to those registers are macros defined in the header dpl.h located under ldk/sources/linux-nps/arch/arc/include/uapi/asm/.
2.	Once a dp process starts to run on a cpu, no other dp processes can run on the same cpu until the previous process has died. In case another process tries to perform ezdp_init_local on a cpu that already runs a dp process, the opening of the dpl module will fail and EBUSY will be stored in errno.
3.	The NPS now supports only one physical function and one lane for the PCI. Therefore, the configuration of the PCI was changed. The U-Boot environment variable pci_config was changed to pci_gen. The number of physical functions and lanes for the PCI cannot be changed, and the value that pci_gen should get is a number between 1 and 3 to indicate the PCI generation currently used by the NPS. For example, changing the PCI to Gen 3 will be done like this (from the u-boot environment): nps# setenv pci_gen 3 nps# saveenv

No.	Description
4.	<p>The PCI and PCIe capabilities have been reduced to the minimum required by the NPS. The capabilities that are necessary:</p> <ul style="list-style-type: none"> • Power Management version 3 (PCI capability with ID of 0x1) • MSI (PCI capability with ID 0x5) • Express Endpoint (PCI capability with ID 0x10) • Advanced Error Reporting (PCIe capability with ID 0x1) • Vendor Specific Information (PCIe capability with ID 0xb) • Secondary PCIe Extended (PCIe capability with ID 0x19) <p>The capabilities that were removed:</p> <ul style="list-style-type: none"> • MSI-X (PCI capability) • Vital Product Data (PCI capability) • Virtual Channel (PCIe capability) • Alternative Routing-ID Interpretation (PCIe capability) • Single Root I/O Virtualization (PCIe capability) • Address Translation Service (PCIe capability) • Latency Tolerance Reporting (PCIe capability)
5.	<p>The logs of the kernel describing the cause of an exception have been improved.</p> <p>When an invalid CMEM access (e.g. beyond private or shared CMEM sizes) is performed, the following information will be printed by the kernel: virtual address that was accessed, sizes of the private and shared CMEM and data-cache sections and more information regarding the limits of this access.</p> <p>When an invalid FMT access is performed (e.g. writing to a read-only memory space), the following information will be printed by the kernel: virtual address of the access, number of the FMT slot and the type of the mapped memory space that was accessed (e.g. x4 cluster code).</p> <p>Note: In the two cases above, a SEGFAULT will be sent to the process that performed the access.</p> <p>When an invalid memory space access is performed (e.g. write to a disabled memory space), the following information will be printed by the kernel: number of the memory space that was accessed, type of the memory space that was accessed (e.g. x1 cluster data) and offset of the access. This is in addition to the printings regarding the error code and transaction code that were added for this case in the 2.1a release.</p> <p>Note: In the case above, the core will finish printing another general information regarding ARC generic and auxiliary registers and then will halt. The information will be available, if not through the application, through the dmesg command from the shell.</p>
6.	<p>The format of defining <code>krn_possible_cpus</code> and <code>krn_present_cpus</code> was changed in order to enable the selection of any group of cpus.</p> <p>Example of the new format: Defining 0-7,16-23,32-39,.....,3064-3071,3072-4095 will be done in the following way : <code>0-3071%16=0-7,3072-4095</code>.</p>
7.	<p>Added EZmodpost tool to assist in identifying EMEM mapped functions/variables used in fast path code that could reduce performance. Located in <code>/tools/EZmodpost</code> and <code>/ldk/toolchain/bin/EZmodpost</code>.</p>

No.	Description
8.	<p>The watchdog timer driver has been added to the NPS.</p> <p>The timer is reset every 3 seconds by default by a background process running on cpu 0.</p> <p>To change the frequency of the reset, one needs to change the configuration file located in <code>ldk/sources/buildroot/package/busybox/Config.in</code> and make busybox.</p> <p>The timeout of the watchdog device may be changed by an ioctl call to the <code>/dev/watchdog</code> device. The timeout ranges from 1 to 5 seconds.</p> <p>The watchdog timer operates on a counter that decreases every cycle.</p> <p>When the watchdog counter reaches 0 the first time, nothing happens, but when it reaches 0 the second time, an interrupt is sent from the watchdog device.</p> <p>The interrupt from the watchdog device reaches the GIM block, which sends it outside the NPS through the INT_OUT line.</p>

5.5 Sample Application Changes and New Features

Table 4: Sample Application Changes and New Features

No.	Description
IPsec Gateway Application	
1.	Refactoring of sample code.
2.	AES_128_GCM_128 optimized to reach 180 G
3.	HMAC_SHA1_96 AES_CBC_128 optimized to reach 145 G
4.	HMAC_SHA256_128 AES_CBC_256 optimized to reach 160 G



Important Note: Existing NPS Application and Target projects may need to be re-generated to conform to the above changes.

6 Bug Fixes

Following are the known issues fixed in this release (since EZdk release 2.1a Open).

6.1 CPE Bug Fixes

Table 5: CPE Fixed Bugs

Index	Description
EZcp	
1.	When initialization of shared CMEM fails, printed error might be wrong.
2.	In the CP library memory partition, a user defined internal memory space with index number 63 was not initialized.
3.	When deleting all entries from a cached hash structure, the entries were not deleted from the NPS memory as well.
EZspy	
4.	Bug fix at print pipe: Wrongfully reported busy threads at cluster due to missing init of counters. "CLUSTER_1: busy threads due to non-posted (read) command (out of 16) = 0x3 (3)."

6.2 LDK Bug Fixes

Table 6: LDK Fixed Bugs

Index	Description
1.	<p>Problematic sequence:</p> <pre>- run demo_dp -run_cpus 16-4095 - killall -15 demo_dp - run demo_dp -run_cpus 16-4095 *** oom killer is invoked ***</pre> <p>Solution:</p> <ol style="list-style-type: none"> 1. Add to kernel parameters, through bootargs environment variable - "slub_max_order=0" , This will overrun the kernel default value 3. 2. Shrink all slabs after each killall: For i in `ls /sys/kernel/slab/*/shrink`; do echo 1 > \$i ; done
2.	Irrelevant values from configuration file were exported to Python and could cause errors in the control plane application.

7 API Changes

Following are the modifications to the CPE APIs and DPE APIs introduced in this release (since EZdk release 2.1a Open).

7.1 API Changes in Rev 18.0300.00

7.1.1 CPE API Changes in Rev 18.0300.00

EZcp

1. EZapiChannel.h

1.1. Updated structs:

- 1.1.1. EZapiChannel_EthRXChannelParams and EZapiChannel_InterlakenRXChannelParams: uiLogicalID valid values are 0-0xFE; removed bFlushErrorFrame, uiPMUErrorQueue and bEnableSequenceNumber.
- 1.1.2. EZapiChannel_LoopbackIFParams: uiLogicalID valid values are 0-0xFE; uiRate valid values are 10312500-103125000.
- 1.1.3. EZapiChannel_TSConfirmationIFParams and EZapiChannel_PMUTimerParams: uiLogicalID valid values are 0-0xFE.
- 1.1.4. EZapiChannel_SystemInfoParams: bEnable default is true.

1.2. Removed commands:

- 1.2.1. EZapiChannel_ConfigCmd_SetPCIeIFParams, EZapiChannel_ConfigCmd_SetPCIeMemSpaceParams, EZapiChannel_ConfigCmd_SetPCIePhysicalFuncParams, EZapiChannel_ConfigCmd_SetPCIeMsgQueueParams, EZapiChannel_ConfigCmd_SetPCIeMsgQueueEventParams, EZapiChannel_ConfigCmd_SetPMUGeneralParams, EZapiChannel_StatCmd_GetPCIeIFParams, EZapiChannel_StatCmd_GetPCIeMemSpaceParams, EZapiChannel_StatCmd_GetPCIePhysicalFuncParams, EZapiChannel_StatCmd_GetPCIeMsgQueueParams, EZapiChannel_StatCmd_GetPCIeMsgQueueEventParams, EZapiChannel_StatCmd_GetPMUGeneralParams, EZapiChannel_StatCmd_GetPCIInfo.

1.3. Removed structs and enums:

- 1.3.1. EZapiChannel_PCIeIFParams, EZapiChannel_PCIeMemSpaceParams, EZapiChannel_PCIePhysicalFuncParams, EZapiChannel_PCIeMsgQueueParams, EZapiChannel_PCIeMsgQueueEvent, EZapiChannel_PCIeMsgQueueEventParams, EZapiChannel_PMUGeneralParams, EZapiChannel_PCIInfo.

2. EZapiIF.h

2.1. Updated structs and enums:

- 2.1.1. EZapiIF_EthMACParams: removed uiCRCOffset; bPadShortFrames if true, all outgoing frames shorter than 64 bytes will be padded to 64 bytes (including CRC) + SWPreamble size; uiMaxFrameLength valid values are 32-12032 and default is 12032.
- 2.1.2. EZapiIF_EthStatCounterId: EZapiIF_EthStatCounterId_RX_FRM_1519_1522/ EZapiIF_EthStatCounterId_TX_FRM_1519_1522 for frames received/transmitted with length 1519-1522 bytes that includes VLAN; new fields EZapiIF_EthStatCounterId_RX_FRM_1519_2047 and EZapiIF_EthStatCounterId_TX_FRM_1519_2047;

- removed EZapiIF_EthStatCounterId_RX_FRM_1523_2047 and EZapiIF_EthStatCounterId_TX_FRM_1523_2047.
- 2.1.3. EZapiIF_EthRXFIFOParams: uiFIFOSize valid values are 128-522240; field prefix changed from “ui” to “ai” in
 aiLowPriorityFCThreshold[EZapiFCU_NUM_OF_PRIORITY_FC_LEVELS],
 aiHighPriorityFCThreshold[EZapiFCU_NUM_OF_PRIORITY_FC_LEVELS],
 aiPriorityDropThreshold[EZapiFCU_NUM_OF_PRIORITY_DROP_LEVELS].
- 2.1.4. EZapiIF_EthTXFIFOParams: uiFIFOSize valid values are 512-573440 in granularity of 512 bytes.
- 2.1.5. EZapiIF_InterlakenRXFIFOParams: uiFIFOSize valid values are 512-522240; field prefix changed from “ui” to “ai” in
 aiLowPriorityFCThreshold[EZapiFCU_NUM_OF_PRIORITY_FC_LEVELS],
 aiHighPriorityFCThreshold[EZapiFCU_NUM_OF_PRIORITY_FC_LEVELS],
 aiPriorityDropThreshold[EZapiFCU_NUM_OF_PRIORITY_DROP_LEVELS];
 uiDiscardThreshold default formula changed.
- 2.1.6. EZapiIF_InterlakenTXFIFOParams: valid values are 512-523776.
- 2.1.7. EZapiIF_OOBFCInCalEntryParams: eMode default is EZapiIF_OOBFCInCalMode_FC_IGNORE; uiStartOffset only valid when Type is 'FCSTATE'.
- 2.1.8. EZapiIF_SyncEthParams: uiSerDes default is 0.
- 2.1.9. EZapiIF_InterlakenInterruptControl: removed bLAIF; struct controls Interlaken / Interlaken_LA interrupt.
3. EZapiFCU.h
- 3.1. Updated structs and enums:
- 3.1.1. EZapiFCU_BudgetIDProfile: field prefix changed from “ui” to “ai” in
 aiPriorityDropThreshold[EZapiFCU_NUM_OF_PRIORITY_DROP_LEVELS].
- 3.1.2. EZapiFCU_IFProfile, EZapiFCU_GroupThresholds and EZapiFCU_GlobalThresholds: field prefix changed from “ui” to “ai” in
 aiLowPriorityFCThreshold[EZapiFCU_NUM_OF_PRIORITY_FC_LEVELS],
 aiHighPriorityFCThreshold[EZapiFCU_NUM_OF_PRIORITY_FC_LEVELS],
 aiPriorityDropThreshold[EZapiFCU_NUM_OF_PRIORITY_DROP_LEVELS].
4. EZapiICU.h
- 4.1. Updated structs and enums:
- 4.1.1. EZapiICU_InterlakenParams: uiRXChannel valid values are 0-63.
- 4.1.2. EZapiICU_L2UserDefinedDestAddrAndEthertype: aucLLCSnap default is 0h0000000000000000; aucLLCSnapMask default is 0hFFFFFFFFFFFFFFFF.
5. EZapiPRM.h
- 5.1. Updated structs and enums:
- 5.1.1. EZapiPrm_DebugFlagID:
 removed EZapiPrm_DebugFlagId_WRITE_STATISTIC_CP_TO_DP and
 EZapiPrm_DebugFlagId_SERDES_PCIE_SKIP_RESET;
 added EZapiPrm_DebugFlagId_PMU_ERROR_QUEUE and
 EZapiPrm_DebugFlagId_INTERLEAVE_WRITING_ENABLE.
6. EZapiShared.h
- 6.1. #define EZapiCP_MAX_STRUCTURES increased from 128 to 256.
7. EZapiStat.h
- 7.1. Updated structs and enums:
- 7.1.1. EZapiStat_ShadowGroupParams: uiGroup valid values 0-6.
- 7.1.2. EZapiStat_DoubleCounter: uiByteValueMSB valid values 0-0x7FFFF;

uiFrameValueMSB valid value 0-0xFFF.

7.2. EZapiStruct.h

7.2.1. Updated structs and enums:

7.2.1.1. EZapiStruct_StructParams: uiResultSize for internal table can be 8, 16, 32 bytes.

7.3. EZapiTCAM.h

7.3.1. Commands removed:

7.3.1.1. EZapiTCAM_ConfigCmd_SetExtTCAMContextBuffer,
EZapiTCAM_StatCmd_GetExtTCAMContextBuffer.

7.3.2. Removed structs and enums:

7.3.2.1. EZapiTCAM_ExtTCAMContextBuffer,

7.3.3. Updated structs and enums:

7.3.3.1. EZapiTCAM_IntTCAMLookupTable: uiNumIndexes default is 0;
uiNumBanks with a duplication ratio larger than 4 is not allowed.

7.4. EZapiTM.h

7.4.1. Updated structs and enums:

7.4.1.1. EZapiTM_WREDLevelParams and EZapiTM_QueueDepth:
uiLevel valid values are 0-4, EZapiTM_LEVEL_GLOBAL.

7.4.1.2. EZapiTM_EntityShapingProfileParams: changes to conditions on ProfileType.

7.4.1.3. EZapiTM_EntityPriorityParams: uiLevel valid values are
0-3, EZapiTM_LEVEL_FAST_QUEUE.

7.4.1.4. EZapiTM_PacketSwitchIDParams: uiTXOQChannel default formula changed.

EZdev

8. EZdev.h

8.1. File restructured.

8.2. Removed typedef EZapiChannel_PCInfo EZdev_PCInfo;

EZenv

9. EZutil.h

9.1. Added EZutil_ABS to get absolute value.

10. EZversion.h

10.1. Updates throughout the file.

EZspy

11. EZspy.h

11.1. New structure: EZspy_SingleMonitorParams is the associated structure for
EZspy_StatCmd_MonitorBasic.

11.2. Updated structs and enums:

11.2.1. EZspy_PrintPipeIF: new fields auiIFIFOFullnessFlit, auiMainFIFOFullnessFlit,
auiOFIFOFullnessFlit.

7.1.2 DPE API Changes in Rev 18.0300.00

EZdp

12. Removed files

12.1. ezdp_age.h, ezdp_age_defs.h, ezdp_pci.h, ezdp_pci_defs.h

13. ezdp_atomic.h

13.1. Note that ezdp_atomic_write64_sum_addr and ezdp_atomic_write64_sum_addr_async are not supported for EMEM.

14. ezdp_math.h

14.1. New command ezdp_hash32_non_pow2.

15. ezdp_search.h

15.1. ezdp_validate_hash_struct_desc: new parameter entry_size.

16. ezdp_time.h

16.1. New command ezdp_get_system_tick_lsb.

17. ezdp_version.h revised.

8 Known Limitations

8.1 CPE Known Limitations

Table 7: CPE Known Limitations

No.	Description
EZcp	
1.	For some combinations of keys, the spread of the hash function used by the hash search structure is not ideal, which can lead to signature pages and result pages exhaustion. Implement a workaround by setting the EZapiStruct_HashMemMngParams uiSigPagePercent and uiResPagePercent values to 100.

8.2 DPE Known Limitations

Table 8: Known Limitations

No.	Description
EZdp	
1.	ezdp_load_data_from_sum_addr(_async) and ezdp_store_data_to_sum_addr(_async) APIs: 256-byte values in cannot be immediate in these APIs and the user should use non-immediate values.
EZframe	
2.	The ezframe_send_to_if and ezframe_send_to_tm APIs fail to update the header offset to be optimized. EZFRAME_DONT_OPTIMIZE_HEADER_OFFSET is not functional.

8.3 LDK Known Limitations

Table 9: LDK Known Limitations

Index	Description
1.	The following items are not supported/operational on the NPS-400 real chip in this release: <ul style="list-style-type: none"> Ethernet: 10/40GBASE-KR, Synchronous Ethernet. GCI devices. Operation of the NPS management LAN at 100 Mbps is not supported (only operational at 1 Gbps).