|  |  |
| --- | --- |
| **Freescale Semiconductor** | Document |
| LDPAA AIOP SERVICE LAYER | Number: AIOPSLRN |
| Release Notes for LDPAA AIOP Service Layer Pre-Alpha v0.3 | Doc. Rev. 0.1 Feb. 25 2014 |

**LDPAA AIOP SERVICE LAYER PRE-ALPHA V0.3**

**Release Notes**

Contents

[1. Overview 2](#_Toc381022845)

[2. Compatibility List 3](#_Toc381022846)

[3. New Features in This Release 3](#_Toc381022847)

[4. Quick Start with this release 3](#_Toc381022848)

[5. Contact Information 4](#_Toc381022849)

[6. Known issues 4](#_Toc381022850)

[6.1 General Limitations 4](#_Toc381022851)

[6.2 Known problems 5](#_Toc381022852)

[6.3 Simulator known issues 5](#_Toc381022853)

[7. Verification 5](#_Toc381022854)

[7.1 Clarification 8](#_Toc381022855)

1. Overview

This document describes the main updates included in the LDPAA AIOP Service Layer Pre-Alpha v0.3.

Pre Alpha testing (i.e. verification of subset of features on the simulator without error cases) has been conducted for functions described below.

Applications should use APIs in: aiopsl/src/include/

All other APIs are internal to the Service Layer and should not be called by applications.

In addition, note that:

* For spinlock, use only:
  1. lock\_spinlock
  2. unlock\_spinlock

All other spinlock functions will be removed in future release.

API may be changed in future releases.

API may be updated in future releases to align with the latest HW specs.

This release can be retrieved from GIT:

GIT repository: ssh://gerrit/ldpaa/aiopsl

GIT tag: **ldpaa-aiop-sl-v0.3**

Please see the aiopsl/docs/AIOPCoreLib\_ChangeLog.txt and aiopsl/docs/AIOP\_ARENA\_ChangeLog.txt files for a detailed list of changes.

The API documentation (.chm) may be downloaded from Compass at: <http://compass.freescale.net/livelink/livelink?func=ll&objId=230824100&objAction=browse&viewType=1>

Please subscribe to the AIOPSREL mailing list to receive future release notifications.

1. Compatibility List

This release is bit accurate with respect to the following docs/tools.

As such, it is suitable for running on RTL/Emulator and/or on the AIOP Simulator.

|  |  |
| --- | --- |
| **Tool/Doc** | **Version** |
| AIOP\_Archdef | 0.7.1 |
| FD\_section | 0.62.2 |
| CTLU\_AIOP\_bg | 0.6 |
| Parser\_Block\_Guide | 3.40 (Maturity level 0.95) |
| AIOP\_Instruction\_Additions | 1.0 |
| Compiler | Build 268 |
| CW for DPAA | 10.0.5 |
| Simulator | DPAA Simulator 0\_5\_2\_0068 |

1. New Features in This Release

The main additions/changes in this version are:

* Added L4 functionality – TCP GSO initial functionality for IPv4 packets with TCP flags cleared.
* Added L4 functionality – TCP GRO initial functionality for IPv4 TCP segments.  
  Aggregation is done according to configurable maximum aggregated segments limit per packet.
* Added IP functionality – IP Fragmentation initial functionality for IPv4 frames.
* Added IP functionality – IP Reassembly initial functionality for IPv4 frames with 1 instance and 2 fragments in order, exclusive mode.
* Added SLAB module.   (This module replaces the Virtual Pools module.)
  + Pool Creation/Deletion
  + Buffer Acquire/Release.

These functions are supported only through SLAB module and not through Virtual Pool.

* Added Command Interface functionality
  + AIOP Client to MC Server
* Pre Alpha verification for functions listed in section 7.

1. Quick Start with this release
2. Install CW 10.0.5 from the following link : <http://compass.freescale.net/livelink/livelink?func=ll&objId=231290331&objAction=browse&viewType=1>
3. Unpack  DPAA Simulator 0\_5\_1\_0068  from: <http://classics.ea.freescale.net/cgi-bin/trans/index.pl?path=/Builds/Components/Simulator/swdsp/DPAA_SIM_RELEASE/>

package: **DPAA\_SIM\_RELEASE\_0\_5\_2\_0068.tgz** on a Linux64-bit host.

1. Pull the aiopsl sources from  aiopsl git repository:
   * 1. git clone ssh://gerrit/ldpaa/aiopsl
     2. Switch to **ldpaa-aiop-sl-v0.3** branch:

git checkout **ldpaa-aiop-sl-v0.3**

1. Navigate to the simulator directory and run the simulator:
   * 1. cd to <path\_to\_simulator>/dtsim\_release/linux64/>
     2. Copy the file: aiopsl\build\aiop\_t4ls\_sim\sim\_files\t4ls\_sys\_test.cfg into <path\_to\_simulator>/dtsim\_release/linux64/
     3. Set environment variables:
        1. SIM\_INIT\_PARAMS\_FILE=lsdpaa2sim\_init\_params.cfg
        2. VIPR\_STARTUP\_PYTHON\_SCRIPTS= <python script file path>
2. Create a new CW workspace and import the aiop related CW projects: “app\_process\_packet” and “aiop\_sl” from <aiopsl\_git\_repository>/build.
3. Build app\_process\_packet application (it builds also aiop\_sl statically linked library).
4. Modify the connection settings, based on the set-up environment (Linux64-bit host machine and ccssim2 port).
5. Debug the app\_process\_packet project.   
   Please refer to the README file in aiopsl\build\aiop\_t4ls\_sim\apps\app\_process\_packet\src\   
   for instructions.
6. The demo requires running the simulator in standalone mode. To set up the standalone mode, update lsdpaa2sim\_init\_params.cfg as follows:
   * 1. Remove ccm.enable\_byte\_swap=false.
     2. Modify section #cores to have only this setting: aiop.num\_cores=1.
     3. Set section aiop.standalone=true inside #aiop.
     4. Remove wrp.boot\_release\_gpp=true inside #wapper.
7. Run the simulator as follows : ./start\_ccssim2 -port 41476 –smodel "ls\_sim\_config\_file=t4ls\_sys\_test.cfg”
8. Contact Information

* Mail List: **AIOPSREL**
* Bug Reporting [Clear Quest](http://cq.freescale.net/cqweb/) BINs: **LS-AIOP-LOW-LEVEL** and **LS-ARENA**

1. Known issues

## General Limitations

* Slab creation does not support additional buffers beyond the committed number (i.e. only extra\_buffs=0 is currently supported).
* The Slab module currently supports buffer sizes up to 256B in PEB and up to 1024B in the DDR.
* BPID 1 must be configured to include at least 1 buffer of size 256 bytes.

This buffer is used for parse profile ID generation. This limitation will be removed in future releases.

* BPID 2 must be configured to include at least 1 buffer of size 1024 bytes.

This buffer is used for key ID generation. This limitation will be removed in future releases.

## Known problems

* The slab\_create() and slab\_free() functions can only be called from the application initialization/shutdown routines (app\_init()/app\_free()).
* The fsl\_os\_print() function runs successfully when called from the application initialization routine (app\_init()), but fails when called from the application runtime routine (app\_process\_packet()).  A workaround will be provided when available.
* The fsl\_os\_malloc() and fsl\_os\_xmalloc() functions run successfully when called from the application initialization routine (app\_init()).  The runtime routine (app\_process\_packet()) can only call fsl\_os\_xmalloc() with partition MEM\_PART\_PEB or MEM\_PART\_SH\_RAM.  A workaround will be provided when available.

## Simulator known issues

* FDMA insert command result a crash in the simulator (ENGR297430).
* Incorrect work of the simulator with 2 frames (ENGR299900).
* The stack overwrite the PA memory space while enlarge the PA size (ENGR00295789).
* The DISC has incorrect value in the simulator print out (ENGR00298955).
* Incorrect “table is full” report (ENGR00298793).
* FDMA replace command with SA field==1 fails (NGR00299014).
* FDMA repalce command error (ENGR300221).
* Parser reports on IPv6 FragmentOffset for an IPv4 frame (ENGR298635).

1. Verification

A Pre Alpha testing has been conducted for the following functions:

|  |
| --- |
| **STE Functions** |
| ste\_set\_32bit\_counter() |
| ste\_set\_64bit\_counter() |
| ste\_inc\_counter() |
| ste\_dec\_counter() |
| ste\_inc\_and\_acc\_counters() |
| ste\_inc\_and\_sub\_counters() |
| ste\_dec\_and\_acc\_counters() |
| ste\_dec\_and\_sub\_counters() |
| ste\_barrier() |
| **FDMA Functions** |
| fdma\_present\_default\_frame() |
| fdma\_present\_frame() |
| fdma\_present\_default\_frame\_segment() |
| fdma\_present\_frame\_segment() |
| fdma\_read\_default\_frame\_asa() |
| fdma\_read\_default\_frame\_pta() |
| fdma\_extend\_default\_segment\_presentation() |
| fdma\_store\_default\_frame\_data() |
| fdma\_store\_frame\_data() |
| fdma\_store\_and\_enqueue\_default\_frame() |
| fdma\_enqueue\_default\_fd() |
| fdma\_discard\_default\_frame() |
| fdma\_terminate\_task() |
| fdma\_concatenate\_frames() |
| fdma\_split\_frame(). |
| fdma\_trim\_default\_segment\_presentation() |
| fdma\_modify\_default\_segment\_data() |
| fdma\_replace\_default\_segment\_data() |
| fdma\_insert\_default\_segment\_data() |
| fdma\_insert\_segment\_data() |
| fdma\_delete\_default\_segment\_data() |
| fdma\_close\_default\_segment() |
| fdma\_replace\_default\_asa\_segment\_data() |
| fdma\_replace\_default\_pta\_segment\_data() |
| fdma\_calculate\_default\_frame\_chekcsum() |
| fdma\_create\_frame() |
| **TLU Functions** |
| ctlu\_table\_create() |
| ctlu\_table\_rule\_create() |
| ctlu\_table\_get\_params() |
| ctlu\_table\_lookup\_keyid() |
| ctlu\_table\_lookup\_key() |
| ctlu\_kcr\_create() |
| ctlu\_kcr\_builder\_init() |
| ctlu\_gen\_key() |
| ctlu\_kcr\_add\_constant\_fec() |
| ctlu\_kcr\_add\_protocol\_specific\_field() |
| ctlu\_kcr\_add\_protocol\_based\_general\_fec() |
| **TMAN Functions** |
| tman\_create\_tmi() |
| tman\_create\_timer() |
| tman\_recharge\_timer() |
| tman\_timer\_completion\_confirmation() |
| tman\_delete\_tmi() |
| tman\_query\_tmi() |
| tman\_get\_timestamp() |
| tman\_delete\_timer() |
| tman\_increase\_timer\_duration() |
| tman\_query\_timer() |
| **CDMA Functions** |
| cdma\_read() |
| cdma\_write () |
| cdma\_refcount\_decrement() |
| cdma\_mutex\_lock\_take() |
| cdma\_mutex\_lock\_release() |
| cdma\_read\_with\_mutex() |
| cdma\_write\_with\_mutex() |
| cdma\_refcount\_increment() |
| cdma\_write\_lock\_dma\_read\_and\_increment() |
| cdma\_write\_release\_lock\_and\_decrement() |
| cdma\_ws\_memory\_init() |
| **Parser Functions** |
| parser\_profile\_create() |
| parser\_result\_generate\_default() |
| **Network Interface** |
| dpni\_drv\_send() |
| **L2 Functions** |
| l2\_header\_remove() |
| l2\_vlan\_header\_remove() |
| l2\_set\_dl\_src() |
| l2\_set\_dl\_dst() |
| **IP Functions** |
| ipv4\_header\_modification() |
| ipv6\_header\_modification() |
| ipv4\_header\_encapsulation() |
| ipv6\_header\_encapsulation() |
| ip\_header\_decapsulation() |
| **IP Functions – IP Fragmentation** |
| ipf\_context\_init() |
| ipf\_generate\_frag() |
| **IP Functions – IP Reassembly** |
| ipr\_create\_instance() |
| ipr\_delete\_instance() |
| ipr\_reassemble() |
| **L4 Functions – TCP GSO** |
| tcp\_gso\_context\_init() |
| tcp\_gso\_generate\_seg() |
| **L4 Functions – TCP GRO** |
| tcp\_gro\_ aggregate\_seg() |
| **NAT Functions** |
| nat\_ipv4() |
| nat\_ipv6() |

|  |
| --- |
| **Memory Management** |
| fsl\_os\_malloc()  fsl\_os\_malloc\_smart() |
| **Console IO** |
| fsl\_os\_print() |
| **Network Utilities** |
| inet\_pton() |
| inet\_ntop() |
| **SLAB** |
| slab\_create() |
| slab\_free() |
| slab\_acquire() |
| slab\_release() |
| slab\_debug\_info\_get() |
| **Command Interface** |
| cmdif\_open() |
| cmdif\_send() |

## Clarification

The following was not verified in this release, will be verified in future releases:

* L4 Function – TCP GSO, not verified :
  + - IPv4 and TCP header options.
    - IPv4 and TCP checksum for new segments.
    - IPv4 ID generation for new segments.
    - IPv6.
    - Error cases.
* L4 Function – TCP GRO, not verified:
* IPv6 segments.
* Timeout operation.
* IPv4 and TCP checksum for the large packet.
* Metadata.
* Aggregation limits:
  + Packet size
  + Timeout
* Statistics counters.
* TCP flags.
  + All TCP flags must be cleared in the incoming segments.
* TCP timestamp
  + All segments should not contain a TCP timestamp.
* IP ECN bits
  + Must be cleared in the incoming segments.
* Unexpected segment sequence number.
* Segment ACK number is less than the ACK number of the previously coalesced segment.
* Error cases
* IP Function - IP Fragmentation, not verified:
* IPv4 options
* IPv6
* Fragmentation of fragments
* Fragments restoration
* Error cases
* IP Function - IP Reassembly, not verified
* IPv4 checksum generation.
* TCP checksum generation.
* No congestion indication support
* Out of order.
* timeout
* IPv6
* Concurrent mode.
* IPR statistics
* Error cases.

***How to Reach Us:***

**Home Page:**

www.freescale.com

**email:**

support@freescale.com

**USA/Europe or Locations Not Listed:**

Freescale Semiconductor

Technical Information Center, CH370

1300 N. Alma School Road

Chandler, Arizona 85224

(800) 521-6274

480-768-2130

support@freescale.com

**Europe, Middle East, and Africa:**

Freescale Halbleiter Deutschland GmbH

Technical Information Center

Schatzbogen 7

81829 Muenchen, Germany

+44 1296 380 456 (English)

+46 8 52200080 (English)

+49 89 92103 559 (German)

+33 1 69 35 48 48 (French)

support@freescale.com

**Japan:**

Freescale Semiconductor Japan Ltd.

Headquarters

ARCO Tower 15F

1-8-1, Shimo-Meguro, Meguro-ku

Tokyo 153-0064, Japan

0120 191014

+81 2666 8080

support.japan@freescale.com

**Asia/Pacific:**

Freescale Semiconductor Hong Kong Ltd.

Technical Information Center

2 Dai King Street

Tai Po Industrial Estate,

Tai Po, N.T., Hong Kong

+800 2666 8080

support.asia@freescale.com

**For Literature Requests Only:**

Freescale Semiconductor

Literature Distribution Center

P.O. Box 5405

Denver, Colorado 80217

(800) 441-2447

303-675-2140

Fax: 303-675-2150

LDCForFreescaleSemiconductor

@hibbertgroup.com

Information in this document is provided solely to enable system and software

implementers to use Freescale Semiconductor products. There are no express or

implied copyright licenses granted hereunder to design or fabricate any integrated

circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to

any products herein. Freescale Semiconductor makes no warranty, representation or

guarantee regarding the suitability of its products for any particular purpose, nor does

Freescale Semiconductor assume any liability arising out of the application or use of

any product or circuit, and specifically disclaims any and all liability, including without

limitation consequential or incidental damages. “Typical” parameters which may be

provided in Freescale Semiconductor data sheets and/or specifications can and do

vary in different applications and actual performance may vary over time. All operating

parameters, including “Typicals” must be validated for each customer application by

customer’s technical experts. Freescale Semiconductor does not convey any license

under its patent rights nor the rights of others. Freescale Semiconductor products are

not designed, intended, or authorized for use as components in systems intended for

surgical implant into the body, or other applications intended to support or sustain life,

or for any other application in which the failure of the Freescale Semiconductor product

could create a situation where personal injury or death may occur. Should Buyer

purchase or use Freescale Semiconductor products for any such unintended or

unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor

and its officers, employees, subsidiaries, affiliates, and distributors harmless against all

claims, costs, damages, and expenses, and reasonable attorney fees arising out of,

directly or indirectly, any claim of personal injury or death associated with such

unintended or unauthorized use, even if such claim alleges that Freescale

Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc.

The described product contains a PowerPC processor core. The PowerPC name is a

trademark of IBM Corp. and used under license. All other product or service names are

the property of their respective owners.

© Freescale Semiconductor, Inc., 2005-2008.