Traffic bifurcation using VSP on LS1046ARDB

This document introduces FMAN VSP hardware overview, the usage of Virtual Storage Profiles, FMan VSP driver and traffic bifurcation using VSP on LS1046ARDB.

1. FMan VSP Hardware Overview

VSPs may be used by user for virtualization. If a user is running with a multi-partitioned (or with a multiple software entities) system where a single MAC may be used by several software partitions/entities simultaneously, except for using a different FQID (that is already available in DPAA1.0), user may use a different VSP for each SW partition/entity; that way, the buffer may be private (rather than being shared as in DPAA1.0). It allows the virtualization of the buffer pool selection for frame storage (and other parameters related to storage in external memory) from the physical hardware ports. Using this mechanism, different packets received on the same physical port may be stored in different BM pools based on the frame header, in a similar way to FQID selection. VSPs are replacing the legacy, "physical", per-port BM Pool selection. A backward compatible mode exists and it is possible to use the original BM Pool selection, now referred to as "Physical SP".

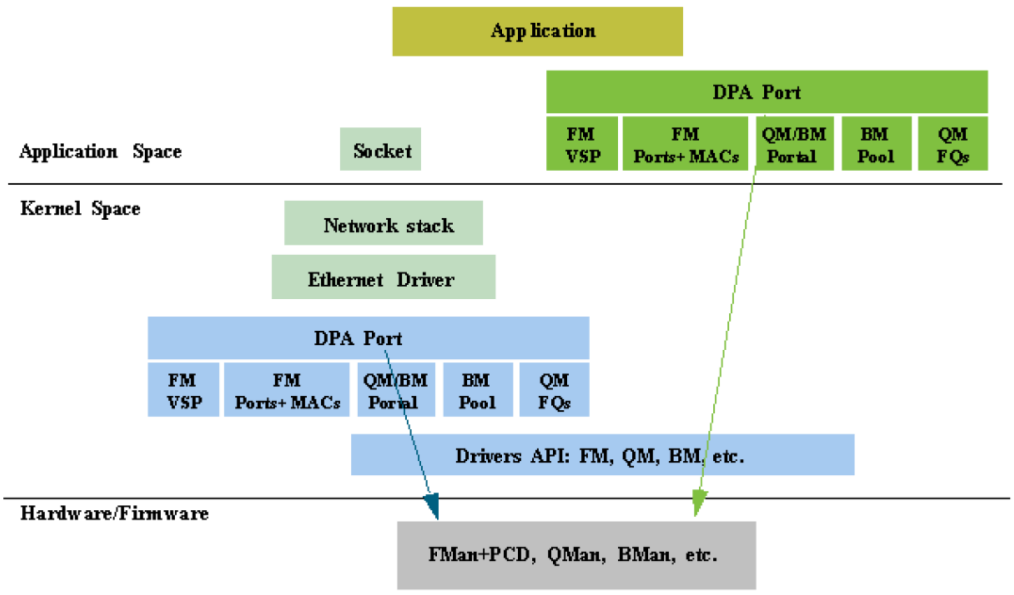


Figure 113. Virtualization Using VSPs

The global FMan module is in charge of the Virtual Storage Profiles entries management. On FMan initialization, the first VSP index dedicated to this partition must be defined (it should be an absolute index), and so is the total number of VSP's for this partition. Later, for each port using VSP's, a window of entries should be defined. VSPs may not be shared among FMan ports. Each port has a default VSP. On each PCD classification, a VSP may be selected. Received packets will be written into the destination buffer according to the VSP parameters, while the VSP is selected according to the frame headers and the PCD configuration. The VSP is implemented by the driver as separate entity, however, other modules of the FM driver are aware of this entity and interact with it. An FM VSP module represents a single storage profile. The global FMan module is in charge of the Virtual Storage Profiles entries management. On FMan port initialization, if using VSP mode, it should allocate and bind to a range of VSP's. On the PCD, A decision is being taken by user on every node of the PCD graph whether to continue to work with previously defined VSP or to override with a new profile.

2. The usage of Virtual Storage Profiles

An FMan Port may use the legacy Physical Storage Profile or the Virtual Storage Profiles (VSP). This section will discuss the usage of VSP by a FMan port.
When a user wants to set an Rx or OP port to work in virtualization mode using VSP's rather than the physical SP, user should call the function which allocates a storage profile window (range of VSPs allocated in continuously manner) to a port. The user should also define which profile in this range should be used as default SP; note that the default profile should be a relative index within the allocated window. Upon calling the window allocation routine, the driver enables virtual mode (that is, using VSPs) for this port, allocates its profiles and defines default SP. In order to redirect a packet into a certain VSP, user may set the 'relative-VSP-id' within the PCD graph nodes (For example, in the match-table entries). The value in the PCD graph nodes is port relative so if two ports are sharing the same PCD graph node (For example, a match-table), the actual VSP will be selected by the 'relative-VSP-id' plus the port's base VSP as shown in the figure below.

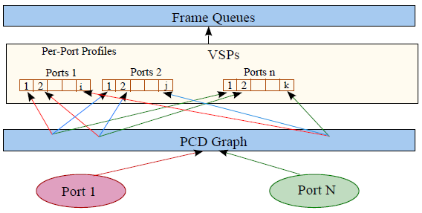


Figure 111. FMan PCD graph and the VSP selection

Rules and restrictions regarding the use of VSP:

- When called for Rx ports, the allocation routine expects also the handle of coupled Tx port as a parameter; the driver sets automatically the Tx port to work in VSP mode also and use the same default profile for this port.
- Storage Profiles windows may not overlap; that is, sharing of VSPs between FM ports is not allowed by the driver.
- A call to the allocation routine requires that the FM port will be disabled. In the case of Rx port, coupled Tx port should also be disabled. When an FM Port (that has VSP mechanism enabled) is enabled, at least the default profile must be initialized.

- A call to the allocation routine may not be reverted, that is, it's impossible to disable virtualization mode.
- Number of profiles to be allocated must be a power-of-2. In addition, the "base-profile" that will be allocated by the driver will be aligned to the number-of-profiles provided by the user.
- For FM-Port that works with VSP, its classification should also use VSP; that is, classification (For example, KG scheme or CC-node) should NOT try to revert from VSP to the FM-Port "physical" SP
- When user frees all resources of FM port, the driver frees automatically VSP window which has been allocated for this port.

3. FMan VSP Driver

The VSP is implemented by the driver as separate entity, however, other modules of the FM driver are aware of this entity and interact with it. An FM VSP module represents a single storage profile. The global FMan module is in charge of the Virtual Storage Profiles entries management. On FMan port initialization, if using VSP mode, it should allocate and bind to a range of VSP's. On the PCD, A decision is being taken by user on every node of the PCD graph whether to continue to work with previously defined VSP or to override with a new profile.
This module represents the FMan VSP driver. It includes:

- FMan VSP hardware structures configuration and enablement
- Parsing of the buffer
- Statistics

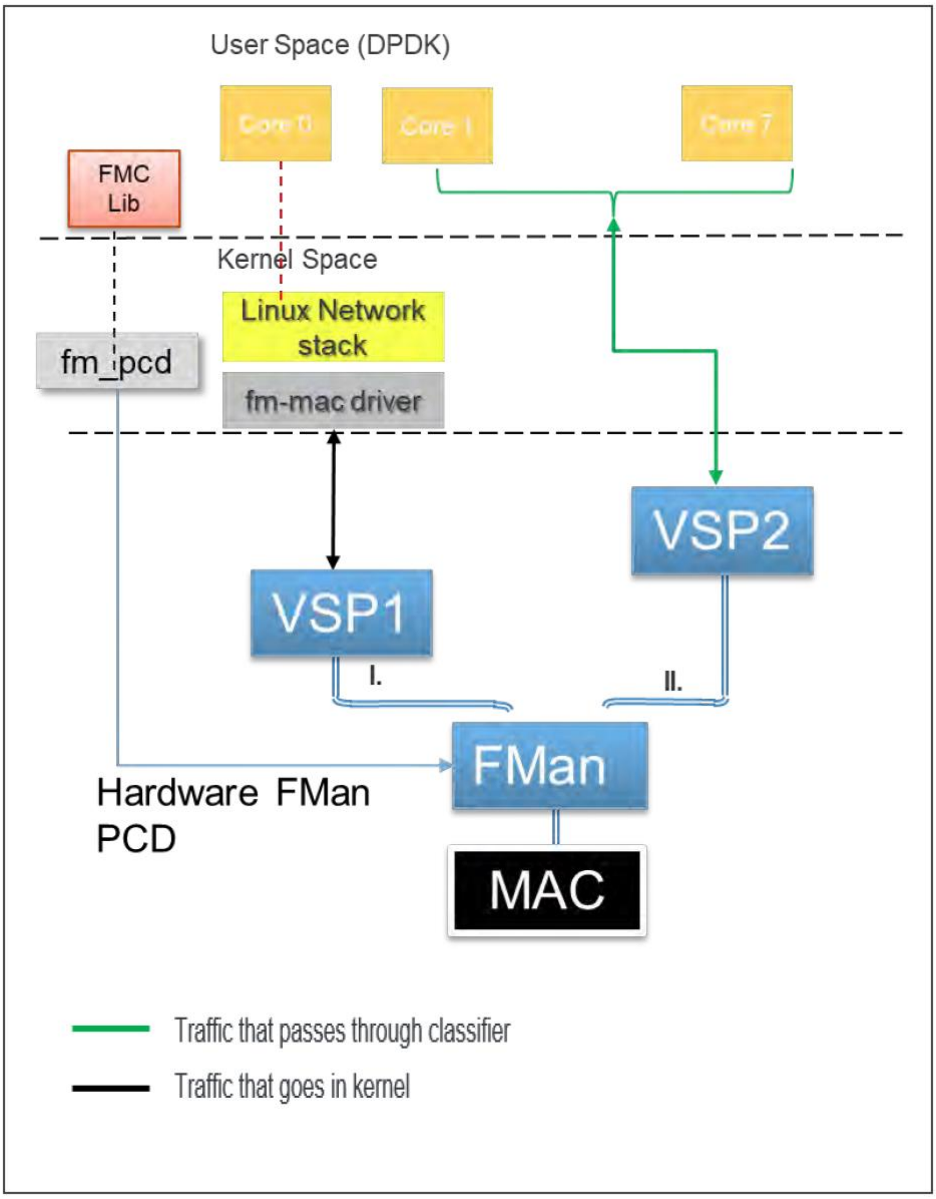
FMan VSP Driver Sequence This sequence includes other modules required for the VSP

- Definition of general VSP parameters on global FMan initialization
- FM Port initialization
- FM Port VSP window allocation
- FM Port enablement
- FMan VSP Config routine (for specific VSP's)
- [Optional] FMan VSP advance configuration routines (for specific VSP's)
- FMan VSP Init routine (for specific VSP's)

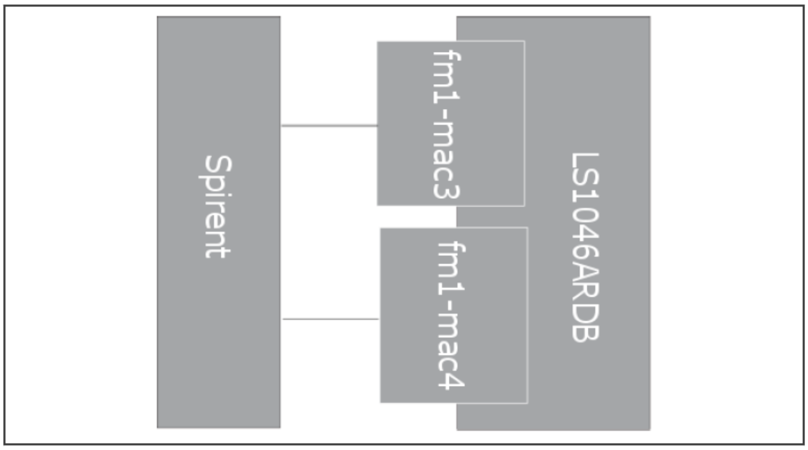
The following sections describe main driver functionalities and their usage.
Virtual Storage Profile Initialization The VSP's must be initialized prior to their usage. It is user's responsibility to initialize at least the default VSP for each port before enabling it. Similarly, it is their responsibility to initialize all other VSPs before a classification that may use some VSP is enabled. Initializing a VSP defines the destination BM Pool buffer for a specific type of packets. It also defines the structure of the buffer - that is, the data offset, the prefix content, and so on.
Virtual Storage Profile Parsing On VSP initialization, the user defines the buffer prefix content. Based on these requirements, the driver then defines the buffer prefix structure, that is, data offset, whether certain information such as parse result should be copied to the external buffer and where it will be located. On buffer reception, the user may call VSP routines in order to get the data, as well as the buffer prefix sections such as parse result, timestamp, or Keygen output.

4. Traffic bifurcation using VSP on LS1046ARDB

DPAA supports Hardware (FMan) based traffic splitting on different interfaces. Custom method to split the traffic can be programmed via FMan PCD interface configurations. Interface can receive packets on different buffer pools (Virtual Storage profile).



This document uses LS1046ARDB for demonstrating the use case.



In the above figure, an NXP LS1046ARDB board is shown connected to a packet generator (Spirent). The example uses Spirent as packet generator, however, any other source of controlled packet transmission can also be used. The figure uses fm1-mac3 and fm1-mac4 interfaces for demonstration.

Implement VSP mode

To activate VSP capability on a port, the user will have to configure the chosen node in the device tree. One valid entry looks like:

```
fman0_rx2-extd-args {
        cell-index = <2>;
        compatible = "fsl,fman-port-1g-rx-extended-args";
        /* Define Virtual storage profile */
        /* <number of profiles, default profile id> */
        vsp-window = <2 0>;
};
```

vsp-window - the number of VSPs that this port can have and the default VSP.

Ethernet port definition is as the following in dts file.

```
ethernet@2 {
        compatible = "fsl,ls1046-dpa-ethernet-shared", "fsl,dpa-ethernet-shared";
        fsl,bman-buffer-pools = <&bp7 &bp8 &bp9>;
        fsl,qman-frame-queues-rx = <0x54 1 0x55 1>;
        fsl,qman-frame-queues-tx = <0x74 1 0x75 1 0x80 1>;
};
```

VSP mode definition in fmc policy file as the following.

```
!--     xxxxxxxxxxxxxxxxxxxxxxxxx Distributions xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->

    <vsp name="shared_mac_pool" base="0"/>
    <vsp name="dpdk_mac3_flows" base="1"/>
```

```
<vsp name="dpdk_mac4_flows" base="1"/>
<vsp name="dpdk_mac5_flows" base="1"/>
<vsp name="dpdk_mac6_flows" base="1"/>
<vsp name="dpdk_mac9_flows" base="1"/>
<vsp name="dpdk_mac10_flows" base="1"/>
      <classification name="dpdk_mac3_classif" masks="yes">
        <key>
                <fieldref name ="ipv4.nextp"/>
        </key>
        <entry>
                <vsp name="dpdk_mac3_flows" />
                <data>0x11</data>
                <mask>0xFF</mask>
                <queue base="0x700" count="1"/>
        </entry>
        <entry>
                <vsp name="dpdk_mac3_flows" />
                <data>0x32</data>
                <mask>0xFF</mask>
                <queue base="0x700" count="1"/>
        </entry>
        <action condition="on-miss" type="distribution" name="default_mac3_dist"/>
</classification>

<distribution name="dpdk_mac3_dist">
        <vsp name="dpdk_mac3_flows" />
        <protocols>
                <protocolref name="ipv4"/>
        </protocols>
        <queue base="0x700" count="1"/>
        <action type="classification" name="dpdk_mac3_classif" />
</distribution>

<distribution name="default_mac3_dist">
        <vsp name="shared_mac_pool" />
        <queue count="1" base="0x55"/>
</distribution>
```

Steps to run VSP mode as the following.

1. Flash the board using LSDK images.

2. Reboot the board and set dtb as fsl-ls1046a-rdb-usdpaa-shared.dtb on the bank from which board is being booted and boot up the board.

3. Cleanup the current fmc configuration:

```
$ fmc -x
```

4. Set the Ethernet ports to be used:

```
$ ifconfig fm1-mac3 <ip address>
$ ifconfig fm1-mac4 <ip address>
```

5. Setup hugepages:

```
$ mkdir /dev/hugepages
$ mount -t hugetlbfs hugetlbfs /dev/hugepages
$ echo 512 > /proc/sys/vm/nr_hugepages
```

6. Setup VSP fmc configuration:

```
$ fmc -c /usr/local/dpdk/dpaa/usdpaa_config_ls1046_shared_24g.xml -p /usr/local/dpdk/dpaa/usdpaa_policy_24g_classif_udp_ipsec_1queue.xml -a
```

7. Run l2fwd application:

```
$ dpdk-l2fwd -c 0x3 -n 1 -- -p 0x3
```

Now DPDK will handle UDP or ESP traffic, and kernel will handle rest of the traffic.
All traffic with Experimental Protocol set in IPv4 header is sent to Linux Kernel network stack and is available on the Ethernet interface (fm1-mac3/4). Applications, such as tcp dump can demonstrate the packets coming in. All other traffic is visible in the packet generator reflected by the l2fwd application.