



## Cavium RVU ATF documentation

Revision Date	Author	Change
11/17/2017	Konrad Adamczyk	Version v1.1
07/24/2018	Konrad Adamczyk	Version v1.2
11/13/2018	Konrad Adamczyk	Version v1.3
05/09/2019	Tomasz Michalec	Version v1.4

<b>1. Overview</b>	<b>3</b>
<b>2. RVU types (fixed)</b>	<b>4</b>
<b>3. RVU PFs for LMACs</b>	<b>6</b>
<b>4. FDT layout expected by ARM-TF</b>	<b>8</b>
4.1 ADMIN_PF	8
4.2 SSO_TIM_PF	9
4.3 NPA_PF	9
4.4 CPT_PF	10
4.5 LMAC_PF	11
<b>5. Layout of MSI-X vectors</b>	<b>13</b>

# 1. Overview

The goal of this document is to give the reader explanation of how the Resource Virtualization Units (RVUs) are initialized in ARM-TF (ARM Trusted Firmware). RVU is new unit present on CN9xxx Cavium SoC family. CN96xx SoC contains 16 different RVU PFs. Each PF supports up to 128 SR-IOV VFs.

HRM defines total number 256 HWVFs (Hardware VFs) which can be distributed across 16 RVU PFs.

In general, number of VFs supported by given RVU PF is defined in FDT. If, for particular RVU node, given property is missing, default number of VFs is set for particular RVU PF. Moreover, same behavior is defined for number of MSI-X vectors for particular PF. Details (for particular PF) are presented in following sections of this document.

ARM-TF allocates contiguous physical memory region for RVU AF/PF and PF/VF mailboxes and for MSI-X vectors. This memory region starts at 16M (0x01000000) and it's size is 40M (0x02800000). The reason why memory region starts at 16M offset is that memory below 16M is allocated for ARM-TF BL31 services (which persist through OS runtime).

Audience of this document is supposed to have experience with:

- Understand RVU architecture for CN96xx (presented in HRM)
- Understand FDT (Flattened Device Tree) concept

References:

- [ATF User Guide] available at <https://github.com/ARM-software/arm-trusted-firmware/blob/master/docs/user-guide.md>

## 2. RVU types (fixed)

ARM-TF software distinguish RVU blocks as follows:

- RVU PF 0 (ADMIN\_PF) – PF which has RVU\_PRIV\_PFX\_CFG[AF\_ENA] bit set.

For ADMIN\_PF, as for PCI configuration, ARM-TF sets:

Register's field	Value
PCC_XXX_ID[DEVID]<7:0>	PCC_DEV_IDL_E::RVU_AF (0x65)
PCCPF_XXX_REV[BCC,SC,PI]	PCC_DEV_IDL_E::RVU's class code (0x020000)
PCCVF_XXX_REV[BCC,SC,PI]	PCC_DEV_IDL_E::RVU's class code (0x020000)
PCC_XXX_SROIV_DEV[VFDEV]<7:0>*	PCC_DEV_IDL_E::RVU_VF (0x64)*

\*if given PF has RVU\_PRIV\_PFX\_CFG[NVF] > 0.

- RVU PF 13 (SSO\_TIM\_PF) – PF which is dedicated for SSO\_TIM usage.

For SSO\_TIM\_PF, as for PCI configuration, ARM-TF sets:

Register's field	Value
PCC_XXX_ID[DEVID]<7:0>	PCC_DEV_IDL_E::SW_RVU_SSO_TIM_PF (0xF9)
PCCPF_XXX_REV[BCC,SC,PI]	PCC_DEV_IDL_E::RVU's class code (0x020000)
PCCVF_XXX_REV[BCC,SC,PI]	PCC_DEV_IDL_E::RVU's class code (0x020000)
PCC_XXX_SROIV_DEV[VFDEV]<7:0>*	PCC_DEV_IDL_E::SW_RVU_SSO_TIM_VF (0xFA)*

\*if given PF has RVU\_PRIV\_PFX\_CFG[NVF] > 0.

- RVU PF 14 (NPA\_PF) – PF which is dedicated for NPA usage.

For NPA\_PF, as for PCI configuration, ARM-TF sets:

Register's field	Value
PCC_XXX_ID[DEVID]<7:0>	PCC_DEV_IDL_E::SW_RVU_NPA_PF (0xFB)
PCCPF_XXX_REV[BCC,SC,PI]	PCC_DEV_IDL_E::RVU's class code (0x020000)
PCCVF_XXX_REV[BCC,SC,PI]	PCC_DEV_IDL_E::RVU's class code (0x020000)
PCC_XXX_SROIV_DEV[VFDEV]<7:0>*	PCC_DEV_IDL_E::SW_RVU_NPA_VF (0xFC)*

\*if given PF has RVU\_PRIV\_PFX\_CFG[NVF] > 0.

- RVU PF 15 (CPT\_PF) – PF which is dedicated for CPT usage. Note that CPT node has RVU\_PRIV\_PFX\_CFG[AF\_ENA] bit set.

For CPT\_PF, as for PCI configuration, ARM-TF sets:

Register's field	Value
PCC_XXX_ID[DEVID]<7:0>	PCC_DEV_IDL_E::SW_RVU_CPT_PF (0xFD)
PCCPF_XXX_REV[BCC,SC,PI]	CPT_CLASS_CODE (0x108000)
PCCVF_XXX_REV[BCC,SC,PI]	CPT_CLASS_CODE (0x108000)
PCC_XXX_SROIV_DEV[VFDEV]<7:0>*	PCC_DEV_IDL_E::SW_RVU_CPT_VF (0xFE)*

\*if given PF has RVU\_PRIV\_PFX\_CFG[NVF] > 0.

All other RVU PFs (PF1-12) are described in the 3. RVU PFs for LMACs section.

### 3. RVU PFs for LMACs

ARM-TF relies on the FDT configuration provided by BDK. ARM-TF assumes that there'll be no more than 4 PHY subnodes of given "cgx@0/1/2" node. RVU PF mapping is defined as one RVU PF for given LMAC.

As CN96xx supports up to 3 CGXs and up to 4 LMACs per CGX, the maximum number of RVU PFs is 12, hence up to 12 RVU PFs will be dedicated to LMACs.

In case that the number of LMACs for all CGXs is less than 12, number of RVU PFs that will be defined for LMACs will be the same as the number of LMACs for all CGXs, as per 1:1 mapping of RVU PF:LMAC.

For those RVU PFs (LMAC\_PFs), as for PCI configuration, ARM-TF sets:

Register's field	Value
PCC_XXX_IDL[DEVID]<7:0>	PCC_DEV_IDL_E::RVU (0x63)
PCCPF_XXX_REV[BCC,SC,PI]	PCC_DEV_IDL_E::RVU's class code (0x020000)
PCCVF_XXX_REV[BCC,SC,PI]	PCC_DEV_IDL_E::RVU's class code (0x020000)
PCC_XXX_SROIV_DEV[VFDEV]<7:0>*	PCC_DEV_IDL_E::RVU_VF (0x64)*

\*if given PF has RVU\_PRIV\_PFX\_CFG[NVF] > 0.

For other RVU PFs (unused for LMACs from range 1-12), those RVU PFs are defined as SSO\_TIM/NPA PFs (see 2. RVU types (fixed)) in a manner that 0.75 of unused RVU PFs will be marked as SSO\_TIM PFs and 0.25 of unused PFs will be marked as NPA PFs. Those RVU PFs (from range 1-12, marked as either SSO\_TIM or NPA PFs) have the same configuration as listed in section 2. RVU types (fixed), but they do not have any configured VFs.

Following equations are used in calculating number of various RVU PFs:

**TOTAL\_SSO\_TIM\_PFS\_NO\_VFS** = 0.75\*(**MAX\_LMAC\_PFS** - **TOTAL\_LMAC\_PFS**)

**TOTAL\_NPA\_PFS\_NO\_VFS** = 0.25\*(**MAX\_LMAC\_PFS** - **TOTAL\_LMAC\_PFS**)

where:

**MAX\_LMAC\_PFS** = 12 (const)

**TOTAL\_LMAC\_PFS** = number of PHY subnodes of "cgx@0/1/2" nodes in FDT provided by BDK to ARM-TF (see section 4. FDT layout expected by ARM-TF)

## Examples:

1. CGX0 has 2 LMACs, CGX1 has 1 LMAC, CGX2 has 1 LMAC

RVU PFs configuration is as follows:

- a. RVU PFs 1-4 are configured as LMAC\_PF.  
From the left RVU PFs ( $12[\text{MAX\_LMAC\_PFS}] - 4[\text{TOTAL\_LMAC\_PFS}] = 8$ ):
- b. RVU PFs 5-10 ( $0.75 * 8[\text{left RVU PFs}] = 6$ ) are configured as SSO\_TIM PFs (with NumVFs = 0)
- c. RVU PFs 11-12 ( $0.25 * 8[\text{left RVU PFs}] = 2$ ) are configured as NPA\_PFs (with NumVFs = 0)
- d. Other RVU PFs (0, 13, 14 and 15) are configured as defined in section 2. RVU types - (fixed).

2. CGX0 has no LMACs, CGX1 has 3 LMACs, CGX2 has 2 LMACs

RVU PFs configuration is as follows:

- a. RVU PFs 1-5 are configured as LMAC\_PF.  
From the left RVU PFs ( $12 - 5 = 7$ ):
- b. RVU PFs 6-10 are configured as SSO\_TIM PFs (with NumVFs = 0)
- c. RVU PFs 11-12 are configured as NPA\_PFs (with NumVFs = 0)
- d. Other RVU PFs (0, 13, 14 and 15) are configured as defined in section 2. RVU types - (fixed).

3. CGX0 has 1 LMAC, CGX1 has 4 LMACs, CGX2 has 4 LMACs

RVU PFs configuration is as follows:

- a. RVU PFs 1-9 are configured as LMAC\_PF.  
From the left RVU PFs ( $12 - 9 = 3$ ):
- b. RVU PFs 10-11 are configured as SSO\_TIM PFs (with NumVFs = 0)
- c. RVU PF 12 is configured as NPA\_PFs (with NumVFs = 0)
- d. Other RVU PFs (0, 13, 14 and 15) are configured as defined in section 2. RVU types - (fixed).

## 4. FDT layout expected by ARM-TF

Besides that ARM-TF configures PCI settings to make it easier for OS to distinguish particular RVU, it gives flexibility to configure:

1. Number of VFs for given RVU PF (via **num-rvu-vfs** property)
2. Number of MSI-X vectors (via **num-msix-vec** property)

via proper format of FDT.

Note that ARM-TF does not remove any of the FDT nodes/properties at runtime. ARM-TF treats FDT as read-only blob.

ARM-TF expects that FDT nodes for configuring particular RVU PF as follows:

1. For group of RVU types fixed, all nodes are expected to be placed as subnodes of *ecam2: pci@848020000000* node.
2. For group of RVU PFs for LMACs, additional properties (**num-rvu-vfs** and **num-msix-vec**) are expected to be placed in PHY (e.g. xfi/sgmii/qsgmii/rxaui etc.) nodes, which are actually subnodes of “*cgx@0/1/2*” nodes.

Please note that if sum of all **num-rvu-vfs** properties presented in runtime FDT exceeds number of hardware VFs (256), ARM-TF will report an issue about this incident and will refuse to configure all RVU blocks.

### 4.1 ADMIN\_PF

To set given number of VFs for ADMIN\_PF (RVU PF0), the following format of node is expected:

```
ecam2: pci@848020000000 {
    (...)
    /* RVU admin - PF0 */
    rvu-admin@0 {
        num-rvu-vfs = <8>;
        num-msix-vec = <133>;
    };
};
```

At boot time (when ARM-TF executes), ARM-TF will parse “**rvu-admin@0**” node properties and program following registers:

Register's field	Value
RVU_PRIV_PF(0)_CFG[NVF]*	From property <b>num-rvu-vfs*</b>
RVU_PRIV_PF(0)_CFG[FIRST_HWVF]	0
RVU_PRIV_PF(0)_MSIX_CFG[PF_X]	Based on property <b>num-msix-vec**</b>



RVU_PRIV_PF(0)_MSIX_CFG[VF_X]	198***
-------------------------------	--------

\*if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT\_AF\_PF0\_VFS (0).

\*\*if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT\_MSIX\_AF (37).

\*\*\*specified by NIX LF MSI-X size (131) plus NPA LF MSI-X size (66) plus number of VF interrupts (1). In future version of RVU ATF, if required, this functionality can be extended to support defining number of MSI-X vectors per VF via FDT.

## 4.2 SSO\_TIM\_PF

To set given number of VFs for SSO\_TIM\_PF (RVU PF13), the following format of node is expected:

```
ecam2: pci@848020000000 {
    (...)
    /* SW_RVU_SSO_TIM - PF13 */
    rvu-sso-tim@0 {
        num-rvu-vfs = <8>;
        num-msix-vec = <128>;
    };
};
```

At boot time (when ARM-TF executes), ARM-TF will parse “**rvu-sso-tim@0**” node properties and program following registers:

Register's field	Value
RVU_PRIV_PF(13)_CFG[NVF]*	From property <b>num-rvu-vfs*</b>
RVU_PRIV_PF(13)_CFG[FIRST_HWVF]	RVU_PRIV_PF(0)_CFG[NVF]
RVU_PRIV_PF(13)_MSIX_CFG[PF_X]	Based on property <b>num-msix-vec**</b> and already defined MSI-X vectors
RVU_PRIV_PF(13)_MSIX_CFG[VF_X]	Based on property <b>num-msix-vec**</b>

\*if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT\_VFS (3).

\*\*if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT\_MSIX\_SW (133).

## 4.3 NPA\_PF

To set given number of VFs for NPA\_PF (RVU PF14), the following format of node is expected:

```

ecam2: pci@848020000000 {
    (...)
    /* SW_RVU_SSO_NPA - PF14 */
    rvu-npa@0 {
        num-rvu-vfs = <8>;
        num-msix-vec = <133>;
    };
};

```

At boot time (when ARM-TF executes), ARM-TF will parse “**rvu-npa@0**” node properties and program following registers:

Register's field	Value
RVU_PRIV_PF(14)_CFG[NVF]*	From property <b>num-rvu-vfs*</b>
RVU_PRIV_PF(14)_CFG[FIRST_HWVF]	RVU_PRIV_PF(0)_CFG[NVF] + RVU_PRIV_PF(13)_CFG[NVF]
RVU_PRIV_PF(14)_MSIX_CFG[PF_X]	Based on property <b>num-msix-vec**</b> and already defined MSI-X vectors
RVU_PRIV_PF(14)_MSIX_CFG[VF_X]	Based on property <b>num-msix-vec**</b>

\*if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT\_VFS (3).

\*\*if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT\_MSIX\_SW (133).

#### 4.4 CPT\_PF

To set given number of VFs for CPT\_PF (RVU PF15), the following format of node is expected:

```

ecam2: pci@848020000000 {
    (...)
    /* SW_RVU_CPT - PF15 */
    rvu-cpt@0 {
        num-rvu-vfs = <64>;
        num-msix-vec = <133>;
    };
};

```

At boot time (when ARM-TF executes), ARM-TF will parse “**rvu-cpt@0**” node properties and program following registers:

Register's field	Value
RVU_PRIV_PF(15)_CFG[NVF]*	From property <b>num-rvu-vfs*</b>

RVU_PRIV_PF(15)_CFG[FIRST_HWVF]	RVU_PRIV_PF(0)_CFG[NVF] + RVU_PRIV_PF(13)_CFG[NVF] + RVU_PRIV_PF(14)_CFG[NVF]
RVU_PRIV_PF(15)_MSIX_CFG[PF_X]	Based on property <b>num-msix-vec**</b> and already defined MSI-X vectors
RVU_PRIV_PF(15)_MSIX_CFG[VF_X]	Based on property <b>num-msix-vec**</b>

\*if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT\_VFS (3).

\*\*if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT\_MSIX\_SW (133).

## 4.5 LMAC\_PF

For group of RVU PFs for LMACs, additional properties (**num-rvu-vfs** and **num-msix-vec**) are expected to be placed in PHY (e.g. xfi/sgmii/qsgmii/rxaui etc.) nodes, which are actually subnodes of “cgx@0/1/2” nodes. When using default num-rvu-vfs and num-msix-vec values it is expected to don't add these additional properties. To set given number of VFs for given LMAC\_PF (RVU PF1-12), the following format of PHY node is expected:

```
&mrml_bridge {
    /*
     * This configuration is just an example of usage num-rvu-vfs.
     * Maximum number of HWVFs is 256, hence sum of all num-rvu-vfs
     * cannot exceed 256.
     * ATF will report an issue at boot time when trying to configure
     * more than 256 HWVFs.
     */
    cgx@0 {
        ethernetA0: xfi@00 {
            (...)
            num-rvu-vfs = <8>;
            num-msix-vec = <210>;
        };
        ethernetA1: xfi@01 {
            (...)
            num-rvu-vfs = <8>;
            num-msix-vec = <210>;
        };
        ethernetA2: xfi@02 {
            (...)
            num-rvu-vfs = <8>;
            num-msix-vec = <210>;
        };
        ethernetA3: xfi@03 {
```

```

(...)
num-rvu-vfs = <8>;
num-msix-vec = <210>;
};
};

```

At boot time (when ARM-TF executes), ARM-TF will parse all subnodes of “**cgx@0**”, “**cgx@1**” and “**cgx@2**” nodes for properties and program following registers:

Register's field	Value
RVU_PRIV_PF(1..12)_CFG[NVF]*	From property <b>num-rvu-vfs</b> *
RVU_PRIV_PF(1..12)_CFG[FIRST_HWVF]	Appropriate to previously configured NVFs
RVU_PRIV_PF(1..12)_MSIX_CFG[PF_X]	Based on property <b>num-msix-vec</b> ** and already defined MSI-X vectors
RVU_PRIV_PF(1..12)_MSIX_CFG[VF_X]	Based on property <b>num-msix-vec</b> **

\*if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT\_VFS (3).

\*\*if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT\_MSIX\_LMAC (210).

If particular PHY subnode does not exist (for instance, **cgx@0** node has only 2 PHY subnodes and both **cgx@1** and **cgx@2** will have 4 PHY subnodes), as per 3. RVU PFs for LMACs:Examples section, RVU PFs 11-12 will be configured as SSO\_TIM:NPA PF in 3:1 proportion, respectively. Those RVU PFs will have 0 configured VFs (fixed).

## 5. Layout of MSI-X vectors

All MSI-X vectors must be placed within continuous block of memory. Hardware limits number of MSI-X in t RVU\_PRIV\_CONST[MAX\_MSIX]. Each MSI-X vector require 16 bytes of memory. CN96xx supports up to 32,768 vectors which gives 512KB of memory required for MSI-X vectors. ATF allocate MSI-X vectors starting from address 0x0320 0000.

Region	Number of MSI-X vectors
PF0 vectors	RVU_PRIV_PF(0)_MSIX_CFG[PF_X]
PF0 VFs vectors	RVU_PRIV_PF(0)_MSIX_CFG[VF_X] * RVU_PRIV_PF(0)_CFG[NVF]
PF1 vectors	RVU_PRIV_PF(1)_MSIX_CFG[PF_X]
PF1 VFs vectors	RVU_PRIV_PF(1)_MSIX_CFG[VF_X] * RVU_PRIV_PF(1)_CFG[NVF]
...	...
PFx vectors	RVU_PRIV_PF(x)_MSIX_CFG[PF_X]
PFx VFs vectors	RVU_PRIV_PF(x)_MSIX_CFG[VF_X] * RVU_PRIV_PF(x)_CFG[NVF]
...	...
PF15 vectors	RVU_PRIV_PF(15)_MSIX_CFG[PF_X]
PF15 VFs vectors	RVU_PRIV_PF(15)_MSIX_CFG[VF_X] * RVU_PRIV_PF(15)_CFG[NVF]