Cavium RVU ATF documentation

Revision Date	Author	Change
11/17/2017	Konrad Adamczyk	Version v1.1
07/24/2018	Konrad Adamczyk	Version v1.2

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. CN93xx 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 CN93xx (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 CN93xx 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_ID[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**[MAX_LMAC_PFS] - **4**[TOTAL_LMAC_PFS] = **8**):
   b. RVU PFs 5-10 (**0.75 * 8**[left RVU PFs] = **6**) are configured as SSO_TIM PFs (with NumVFs = 0)
   c. RVU PFs 11-12 (**0.25 * 8**[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 **num-rvu-vfs***
RVU_PRIV_PF(0)_CFG[FIRST_HWVF]	0
RVU_PRIV_PF(0)_MSIX_CFG[PF_X]	Based on property **num-msix-vec****
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 **num-rvu-vfs***
RVU_PRIV_PF(13)_CFG[FIRST_HWVF]	RVU_PRIV_PF(0)_CFG[NVF]
RVU_PRIV_PF(13)_MSIX_CFG[PF_X]	Based on property **num-msix-vec**** and already defined MSI-X vectors
RVU_PRIV_PF(13)_MSIX_CFG[VF_X]	5***

*if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT_VFS (8).
**if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT_MSIX_SW (128).
***specified by TIM LF MSI-X size (2) plus SSO LF MSI-X size (1) plus SSOW LF MSI-X size (1) 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.

4.3 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 **num-rvu-vfs***
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 **num-msix-vec**** and already defined MSI-X vectors
RVU_PRIV_PF(14)_MSIX_CFG[VF_X]	67***

*if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT_VFS (8).
**if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT_MSIX_SW (133).
***specified by NPA LF MSI-X (66) size 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.

4.4 CPT_PF

To set given number of VFs forCPT_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 **num-rvu-vfs***
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 **num-msix-vec**** and already defined MSI-X vectors
RVU_PRIV_PF(15)_MSIX_CFG[VF_X]	129***

*if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT_VFS (8).
**if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT_MSIX_SW (133).
***specified by multiplication of number of CPT LFs (64) and CPT LF MSI-X size (2) 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.

4.5 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. 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 {
                xfi00 {
                        (...)
                        num-rvu-vfs = <8>;
                        num-msix-vec = <210>;
                };
                xfi01 {
                        (...)
                        num-rvu-vfs = <8>;
                        num-msix-vec = <210>;
                };
                xfi02 {
                        (...)
                        num-rvu-vfs = <8>;
                        num-msix-vec = <210>;
                };
                xfi03 {
                        (...)
                        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 **num-rvu-vfs***
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 **num-msix-vec**** and already defined MSI-X vectors
RVU_PRIV_PF(1..12)_MSIX_CFG[VF_X]	198***

*if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT_VFS (8).
**if given property does not exist, ARM-TF will set it to default value, which is specified by DEFAULT_MSIX_LMAC (210).
***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.

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).