Remoteproc VirtIO device restructuring proposal

Arnaud Pouliquen

Objectives

- Create a virtio platform device
- Declare virtio device in device tree
- Prepare the enhancement of the VirtIO services (e.g. use of more than 2 vrings)
- **Keep the legacy working!**

```
&m4_rproc {
    memory-region = <&retram>, <&mcuram>,
                    <&mcuram2>, <&vdev0vring0>,
                    <&vdev0vring1>, <&vdev0buffer>;
    mboxes = <&ipcc 0>, <&ipcc 1>,
             <&ipcc 2>, <&ipcc 3>;
    mbox-names = "vq0", "vq1", "shutdown", "detach";
    status = "okay";
};
```

```
&m4_rproc {
        memory-region = <&retram>, <&mcuram>,
                        <&mcuram2>;
        mboxes = <&ipcc 2>, <&ipcc 3>;
        mbox-names = "shutdown", "detach";
        status = "okay";

        #address-cells = <1>;
        #size-cells = <0>;

        vdev@0 {
                compatible = "rproc-virtio";
                reg = <0>;
                virtio,id = <7>;
                memory-region = <&vdev0vring0>,
                                <&vdev0vring1>, <&vdev0buffer>;
                mboxes = <&ipcc 0>, <&ipcc 1>;
                mbox-names = "vq0", "vq1";
                status = "okay";
        };
};
```

life.augmented

Current situation

- vdev is detected in resource table => virtual remoteproc device is created
  - DMA pool is associated to this device for RPMsg buffer management
    - pool declared in remoteproc platform DT node : vdevXvring0, vdevXvring1, vdevXbuffer
    - or default memory pool: dma_declare_coherent_memory (deprecated)
  - DMA attributes inherited by copy from platform device.
- Concerns from Christoph Hellwing [1] [2]
  - Abuse of the dma_declare_coherent_memory usage and dma_range_map copy (copy_dma_range_map),
  - Use of a virtual remoteproc « fake device »

[1] https://lkml.org/lkml/2021/6/23/607
[2] https://patchwork.kernel.org/project/linux-remoteproc/patch/AOKowLclCbOCKxyiJ71WeNyuAAj2q8EUtxrXbyky5E@cp7-web-042.plabs.ch/

Proposed implementation steps

- step 1: redefine the remoteproc VirtIO device as a platform device
  - migrate rvdev management in remoteproc virtio.c,
  - create a remotproc virtio config (can be disabled in a next step for platform that not use VirtIO IPC).

  => https://github.com/arnopo/linux/commits/step1-virtio-as-pdev
- step 2: add possibility to declare and probe a VirtIO sub node
  - VirtIO bindings declaration,
  - multi DT VirtIO devices support,
  - introduction of a remoteproc virtio device bind mechanism ,

  => https://github.com/arnopo/linux/commits/step2-virtio-in-DT
- step 3: add memory declaration in VirtIO subnode

  => https://github.com/arnopo/linux/commits/step3-virtio-memories
- step 4: add mailbox declaration in VirtIO subnode

  => https://github.com/arnopo/linux/commits/step4-virtio-mailboxes

Use cases to support with DT description:

- Be able to define one reserved memory pool per virtio

  ⇒similar to TI use case

  ⇒Declare the pool in the vdev subnode instead of as the first index of the Remoteproc node

- Be able to have only one global pool for a remote proc

  => does it make sense?

- One mailbox for all vrings notifications

  TBC, but still possible to use kick ops and manage one mailbox for all vrings in the remoteproc node.