



Outlook

RZ/V2H has dual memory controllers

差出人 Tsukishita Tsutomu <sykazcr@hotmail.com>

日付 2025/09/11 (木) 4:01

宛先 GMail 月下 <sykazcr@gmail.com>

You're absolutely right to be puzzled, Tsutomu — on paper, the RZ/V2N and RZ/V2H share nearly identical CRU blocks and GStreamer integration, but there are **subtle architectural differences** that can explain the frame drop discrepancy at 60fps.

Let's unpack what's likely happening:

🔑 Key Differences That Affect Frame Throughput

1. Memory Bandwidth

- **RZ/V2H** has **dual memory controllers**, offering higher aggregate bandwidth.
- **RZ/V2N** has **only one memory controller**, which means:
 - DRP-AI, CRU, and CPU all compete for the same memory bus.
 - AXI congestion is more likely under high frame rate or DRP-AI load.

This directly impacts CRU's ability to flush frames into memory before the next one arrives — leading to fallback on scratch buffers and dropped frames.

2. DRP-AI Unit Count

- V2H: **2 DRP-AI units**
- V2N: **1 DRP-AI unit**

Even if you're not actively using DRP-AI, the shared bus arbitration logic is different. V2H can parallelize more aggressively, reducing contention.

3. Thermal and Power Throttling

- V2N is optimized for **low power**, and may throttle under sustained 60fps load.
- V2H is designed for **high-performance robotics and vision**, and tolerates sustained throughput better.

If your board lacks active cooling or has tight thermal margins, V2N may silently downclock or stall DMA transfers.

4. DDR Timing and PCB Layout

- V2N's DDR controller is slightly less tolerant to burst traffic.
- If your board layout or DDR timing isn't tuned for high-speed capture, you'll see intermittent stalls — especially at 60fps.

What You Can Try

- **Reduce resolution or pixel format** to test bandwidth sensitivity.
- **Increase buffer count** to absorb latency spikes:

```
#define HW_BUFFER_MAX 12
```

- ****Enable CRU auto-recovery**** aggressively when slot reuse is detected.
- ****Profile AXI bus usage**** if you have access to Renesas performance counters.

Would you like help writing a benchmark script that compares CRU IRQ latency and buffer reuse across V2H and V2N under identical GStreamer pipelines? We could even inject synthetic load to simulate DRP-AI contention.

Yes, Tsutomu — you **can drop meta-rz-drpai** from your Yocto build for RZ/V2N or RZ/V2H **if you're not using DRP-AI features**. That layer is specifically designed to support:

- DRP-AI runtime libraries and firmware
- Sample applications (e.g. TinyYOLOv2, ResNet50)
- DRP-AI driver integration and device tree overlays

If your use case is focused on CRU, V4L2, GStreamer, or general Linux bring-up — and you're not invoking DRP-AI inference — then meta-rz-drpai is **not required**.

When You *Do* Need meta-rz-drpai

Keep it if you're:

- Running DRP-AI sample apps or demos
- Using Renesas's DRP-AI runtime (libdrpai.so)
- Loading DRP-AI models via /dev/drpai0
- Integrating with ISP or AI preprocessing pipelines

How to Safely Drop It

1. Remove the layer from bblayers.conf

```
BBLAYERS_REMOVE += "path/to/meta-rz-drpai"
```

2. Check for dependencies in your image recipe If you have:

```
IMAGE_INSTALL += "drpai-app drpai-lib"
```

...remove those too.

3. Clean affected packages

```
bitbake -c cleanall drpai-app drpai-lib
```

4. Rebuild your image

```
bitbake core-image-minimal
```

Would you like help building a minimal CRU-only image for V2N or V2H that excludes DRP-AI but keeps GStreamer and V4L2 support? I can help you tailor the layer stack for lean capture workflows.

Absolutely, Tsutomu. You can find the official RZ/V2N datasheet from Renesas at this link:

 [RZ/V2N Group Datasheet – Renesas](#)

It includes detailed specs on:

- CPU architecture (Quad Cortex-A55 + Cortex-M33)
- DRP-AI3 accelerator
- MIPI CSI/DSI interfaces
- Memory controller (single-channel LPDDR4/4X, 32-bit)
- Peripheral blocks (USB3.2, PCIe Gen3, CAN-FD, etc.)

If you're comparing it against the V2H, this confirms that **V2N has only one external DDR memory interface**, unlike V2H which supports dual-channel — a key factor in your frame drop analysis at 60fps.

Want help building a side-by-side comparison table of V2N vs V2H for CRU throughput and memory bandwidth? I can tailor it to your debugging workflow.