VDOA Firmware Guide

Overview

VDOA is known as Video Data Order Adapter, it is an DMA to convert the data output from VPU to the conventional format that IPU can accept. VDOA is located in 021E4000h in the system memory map.

VPU can output data in tiled mode in order to increase the decoding performance, but the data layout in tiled mode cannot be post processed by the IPU. VDOA is used for the data conversion purpose.

* 1. Features summary

VDOA supports the following features:

* Data conversion from tiled to raster mode
* High resolution support, frame size up to 8192\*4096 pixels
* High speed, working at 264MHz, the burst size is 64bits. In YUV420 partial interleaved mode, 3 pixels can be transferred which means the peak conversion speed is 3\*264=792MP.
* Can support data conversion up to 3 buffers concurrently
  1. Modes of operation

VDOA can support two modes conversion: Sync mode or None Sync mode.

In Sync mode, the VDOA will have handshake with the IPU, after a band of data to be transferred, and to notify IPU to carry on its work. Currently band mode is not supported in IPU driver.

In none Sync mode, the VDOA will work on frame based. The output of VPU is fixed to be YUV420 partial interleaved mode, which is served as the input of VDOA. The input of IPU could be YUV422 interleaved mode or YUV420 partial interleaved mode, which is output of VDOA.

* 1. Clocks

VDOA root clock is derived from the AXI clock, and the frequency is 264MHz as default.

* 1. Resets and Interrupts

VDOA has no reset mechanism.

VDOA can notifiy the CPU when a tranfer is complete. The interrupt ID is 50.

* 1. Initializing the driver

The initialization of VDOA is done in vdoa\_setup(). In this function, the following parameters will be set:

* Frame width and height

* Stride line of VPU output, which is also the input of VDOA
* Stride line of IPU input, which is also the output of VDOA
* Interlaced mode selection. If the VPU output is interlaced, 3 buffers will be used for tranfering the sequential 3 fields of the stream for deinterlaced purpose; or else, only one frame buffer will be used
* Band mode settings. If the VDOA works in sync mode, the driver need to set the band size, and with which IPU the handshake should be done. In non-sync mode, these two parameters are ignored.
* Pixel format selection. This is for the VDOA output, in YUV420 partial interleaved mode or in YUV422 interleaved mode. It will be used further in the IPU post processing.
  1. Testing the driver

The VDOA is tested inside the VPU decoding test. When the map type of VPU is set to be TILED\_FRAME\_MB\_RASTER\_MAP, the VDOA has to be enabled.

Before starting a new VDOA tranfer, the VDOA must be in idle state. Pay attention that the input and output address are all 3LSB aligned.

when running VPU decoding test, there is a prompt to select if VDOA should be enabled. Put ‘Y’ or ‘y’ to confirm, then VDOA will be enabled.

* 1. Source code structure

The code structure is as below:

|  |  |
| --- | --- |
| Description | Location |
| Low-level driver source | ./src/sdk/vdoa/drv/vdoa.c |
| Header files | ./src/sdk/vdoa/inc/vdoa.h |