HiMPP

FAQs

Issue **02**

Date **2013-06-21**

HiSilicon Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.hisilicon.com

Email: support@hisilicon.com

About This Document

Purpose

This document describes the solutions to the problems that may occur when you use the HiSilicon media processing platform (HiMPP).

Related Versions

The following table lists the product versions related to this document.

Product Name	Version
Hi3531	V100
Hi3532	V100
Hi3521	V100
Hi3520A	V100
Hi3520D	V100
Hi3515A	V100
Hi3515C	V100

Intended Audience

This document is intended for:

- Technical support personnel
- Software development engineers

Conventions

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
 DANGER	Indicates a hazard with a high level of risk which, if not avoided, will result in death or serious injury.
 WARNING	Indicates a hazard with a medium or low level of risk that, if not avoided, could result in minor or moderate injury.
 CAUTION	Indicates a potentially hazardous situation, which if not avoided, could result in equipment damage, data loss, performance degradation, or unexpected results.
TIP	Indicates a tip that may help you solve a problem or save time.
 NOTE	Provides additional information to emphasize or supplement important points of the main text.

General Conventions

The general conventions that may be found in this document are defined as follows.

Convention	Description
Times New Roman	Normal paragraphs are in Times New Roman.
Boldface	Names of files, directories, folders, and users are in **boldface**. For example, log in as user **root**.
Italic	Book titles are in *italics*.
`Courier New`	Examples of information displayed on the screen are in `Courier New`.

Change History

Changes between document issues are cumulative. Therefore, the latest document issue contains all changes made in previous issues.

Issue 02 (2013-06-21)

This issue is the second official release, which incorporates the following changes:

Chapter 1 FAQs

In section 1.3, the descriptions are updated.

Section 1.4 is added.

Issue 01 (2012-09-20)

This issue is the first official release, which incorporates the following changes:

Chapter 1 FAQs

In section 1.2, the descriptions of video output and examples are updated.

Issue 00B02 (2012-08-09)

This issue is the second draft release, which incorporates the following changes:

Chapter 1 FAQs

Section 1.2 is added.

Issue 00B01 (2012-06-30)

This issue is the first draft release.

Contents

Figures

1 FAQs

1.1 How to View HiMPP Logs

[Question]

How to view HiMPP logs and change the log level?

[Analysis]

The logs record the error causes, error locations, and system running status during the running of the software development kit (SDK). Logs can help you locate errors.

Currently, the logs are classified into seven levels, and the default level is level 3. A higher log level indicates that more information is recorded. When the level is set to level 7, the information about the running status of the entire system is recorded in logs in real time. The mass information, however, significantly reduces the overall performance of the system. Typically, you are advised to set the log level to level 3. In this case, information is recorded in logs only when errors occur and most errors can be located.

[Solution]

You can run the following commands to obtain logs, and view or change the log level:

- To view the log level of each module, run the **cat /proc/umap/log** command. Then, the log levels of all the modules are listed.
- To change the log level of a module, run the **echo "venc=4" > /proc/umap/log** command. In this command, **venc** is a module name. This name must be the same as that displayed after the **cat** command is executed.
- To change the log levels of all the modules, run the **echo "all=4" > /proc/umap/log** command.
- To obtain logs, run the **cat /dev/log** command. Then, all the log information is displayed. If all the log information is read, the command is blocked until new log information is recorded. Press **Ctrl**+**C** to exit. To use the device node in **/dev/log**, run **open** and **read** commands.

1.2 How to Adjust the Memories Occupied by Media Services

[Question]

Media services require memories for normal running. The memories mainly indicate the media memory zone (MMZ). The HiMPP allocates memories based on services. When the memories are insufficient, you can adjust the allocated memories.

[Analysis]

The HiSilicon SDK allows you to adjust the allocated memories when the memories are insufficient.

[Solution]

Check the operating system (OS) memory and MMZ memory. For details, see chapter 6 "Allocating and Using the Address Space" in the *Description of the Installation and Upgrade of the Hi35xx SDK* under **SDK\01.software\board\documents_cn**. The following describes how to adjust the memories occupied by SDK services by module:

- Video output (VO)
  - [Description]

    1. The number of display buffers (**DispBufLen**) for the VO high-definition (HD) display device is set by calling media processing platform programming interfaces (MPIs).

    The video output unit (VOU) allocates the memory by using the following formula: Device solution x DispBufLen x 1.5 (for semi-planar420 pictures) or 2 (for semi-planar422 pictures)

    2. When the frame-rate-double mode is enabled, the number of display buffers automatically doubles. The default number of display buffers for the HD device is 4 for the Hi3521, Hi3520A, Hi3520D, Hi3515A or Hi3515C, and is 6 for the Hi3531 or Hi3532.

    3. The canvas size of the PIP layer for the HD device can be set to the displayed picture size. That is, the canvas size can be different from the display area size. The VOU does not automatically zoom in on the PIP layer.

    4. When the automatic zoom-in function of the VOU is used, the required memory is reduced.4. The VOU allocates the memory based on the canvas size (**stImageSize**). The canvas size for the HD device can be smaller than the display area size. The VOU automatically zooms in on the video layer to fit into the display area size. This reduces the occupied memory and bandwidth.
  - [Related MPIs] HI_S32 HI_MPI_VO_SetDispBufLen(VO_DEV VoDev, HI_U32 u32BufLen), HI_S32 HI_MPI_VO_GetDispBufLen(VO_DEV VoDev, HI_U32 *pu32BufLen), HI_S32 HI_MPI_VO_SetChnDispPos(VO_DEV VoDev, VO_CHN VoChn, const POINT_S *pstDispPos), HI_S32 HI_MPI_VO_GetChnDispPos(VO_DEV VoDev, VO_CHN VoChn, POINT_S *pstDispPos)
  - [Example]

    1. If **DispBufLen** is changed from 6 to 4 when the display resolution of the HD device is 1080p (semi-planar420 format) and the frame-rate-double mode is enabled, the required memory is reduced by 12 MB.

    2. Assume that the HD display resolution is 1080p, the total size of the display area at the PIP layer is D1 (pictures from multiple channels can be displayed), the canvas size (**stImageSize**) of the PIP layer is D1, and the display area size is 1080p. In this case, the memory required by the PIP layer is D1. You can call HI_MPI_VO_SetChnDispPos to set the display positions of PIP layer channels. Note that the positions must be within the display area.

3. Assume that the HD display resolution is 1080p (semi-planar420 format), the frame-rate-double mode is enabled, and **DispBufLen** is set to **4**. If the automatic zoom-in function of the VOU is used and the canvas size is set to 1280 x 720, the memory is reduced by 13 MB.

- Video decoding (VDEC)
  - [Description] The number of buffers (**DisplayFrameNum**) for storing the output pictures of VDEC channels is set by calling MPIs. **DisplayFrameNum** ranges from 1 to 16, and its default value is 2 in the SDK.
  - [Related MPIs] HI_S32 HI_MPI_VDEC_SetPrtclParam(VDEC_CHN VdChn, VDEC_PRTCL_PARAM_S* pstPrtclParam), HI_S32 HI_MPI_VDEC_GetPrtclParam(VDEC_CHN VdChn, VDEC_PRTCL_PARAM_S* pstPrtclParam)
  - [Example] For a VDEC channel that supports the maximum resolution D1 (semi-planar420 format), if **DisplayFrameNum** is changed from 2 to 1, the required memory is reduced by 600 KB.
- Video process subsystem (VPSS)
  - [Description] When decoded frames are being played, unnecessary functions of the VPSS group that is bound to the decoding channel can be disabled to reduce the memory. When you create a VPSS group, set **bNrEn** to **HI_FALSE**, **enDieMode** to **VPSS_DIE_MODE_NODIE**, and **bHistEn** to **HI_FALSE**. The memory reduced for each VPSS group is calculated as follows: Picture width x Picture height x 2 x 1.5 or 2

    Note: Do not dynamically change settings after the preceding settings are performed. The memory is automatically allocated each time a VPSS group is enabled until the VPSS group is destroyed.
  - [Related MPIs] HI_S32 HI_MPI_VPSS_CreatGrp(VPSS_GRP VpssGrp, VPSS_GRP_ATTR_S *pstGrpAttr)
  - [Example] If the preceding settings are performed for a VPSS group that supports the maximum resolution D1 (semi-planar420 format), the memory is reduced by 1.2 MB.
- Video encoding (VENC)
  - [Description] When a VDEC channel is created, the required buffer is allocated based on the minimum number of reference frames. If you have configured the frame skipping reference mode, the SDK automatically allocates two more reference frame buffers. This ensures that the frame skipping reference service runs properly. Therefore, when the frame skipping reference service is running, more buffers are required.

    Note: If the multi-frame skipping reference mode is set, the allocated buffers are reserved until the VDEC channel is destroyed.
  - [Related MPIs] HI_S32 HI_MPI_VENC_SetH264eRefMode(VENC_CHN VeChn, VENC_ATTR_H264_REF_MODE_E enRefMode)
  - [Example] For a VENC channel that supports the maximum resolution D1 (semi-planar420 format), the required reference frame memory is 1.2 MB by default. If the 2x or 4x frame skipping frame reference mode is set, the required reference frame memory is 2.4 MB for each VENC channel.
- Video detection analysis (VDA)
  - [Description] As the VDA must reserve a reference frame, CIF pictures are recommended for the VDA. If D1 pictures are used, more D1 video buffers (VBs) are occupied.

  - [Example] If the VDA uses 16-channel CIF pictures, only 16 CIF VBs are occupied. If the VDA uses 16-channel D1 pictures, 16 D1 VBs are occupied.
- Video compress (VCMP)
  - [Description] The VCMP function of the graphics layer is disabled by default. To enable the VCMP function, call MPIs. The required memory is calculated as follows: VO display resolution x u32VcmpBufNum x 9/8 x 1.5 or 2**u32VcmpBufNum** is specified when the HiFB driver is loaded, and its value must be 2 or 3.
  - Example: If the VCMP function of a 1080p graphics layer is enabled and **u32VcmpBufNum** is **3**, at most 10 MB memory is required.
- Entire system
  - [Description] Ensure that the sizes of all D1-series pictures can be divided by each other and that the results are integral multiples. For example, a D1 picture size is 704 x 576, a 2CIF picture size is 352 x 576, and a CIF picture size is 352 x 288. If a 2CIF picture size is 352 x 576 and a CIF picture size is 360 x 288, the picture sizes do not meet requirements. In addition, if the size of the picture captured by a VI channel is 720 x 576, the size of the pictured encoded by a VENC channel cannot be 704 x 576.
- Load scripts
  - [Description] When the MMZ driver is loaded by loading scripts, the memory must be allocated to the JPEGD module. The memory is allocated based on the maximum size (1080p) of the JPEG picture to be decoded by default. You can adjust the memory after calculating the required memory as follows: Width of the maximum JPEG picture to be decoded x Height of the maximum JPEG picture to be decoded x 2
  - [Example] Only 810 KB MMZ memory is required to decode 1-channel D1 JPEG pictures.

1.3 How to Improve the VGA Pictures Quality

[Question]

The VGA or HDMI display effect is not satisfied. For example, the luminance, contrast, hue, and saturation of the displayed video do not reach the expected values. In this case, what do I do to optimize the display effect?

[Analysis]

The quality of VGA pictures is related to the VGA circuits. The SDK configurations are tuned based on reference VGA circuits of HiSilicon. If the VGA picture quality is poor, check the VGA circuits of the board.

- If your VGA circuits are the same as the reference VGA circuits delivered by HiSilicon and the VGA configurations for the SDK are default ones, the VGA color bars indicator meets the VGA test standard defined by Tektronix. See Figure 1-2.

  If the VGA display effect is not satisfied (for example, pictures are too bright or the contrast is too large), do as follows:
  - Check the monitor settings.
  - Separately adjust the VGA effect by calling HI_MPI_VO_SetVgaParam for the Hi3531, Hi3532, Hi3521, or Hi3520A or by calling HI_MPI_VO_SetDevCSC() for the Hi3520D, Hi3515A, or Hi3515C. Note that HI_MPI_VO_SetVgaParam does not apply to the Hi3515A or Hi3520D. The related parameters are as follows:

u32Gain: current gain. If pictures are too dark or too bright, you are advised to set this parameter first.

u32luminance: luminance

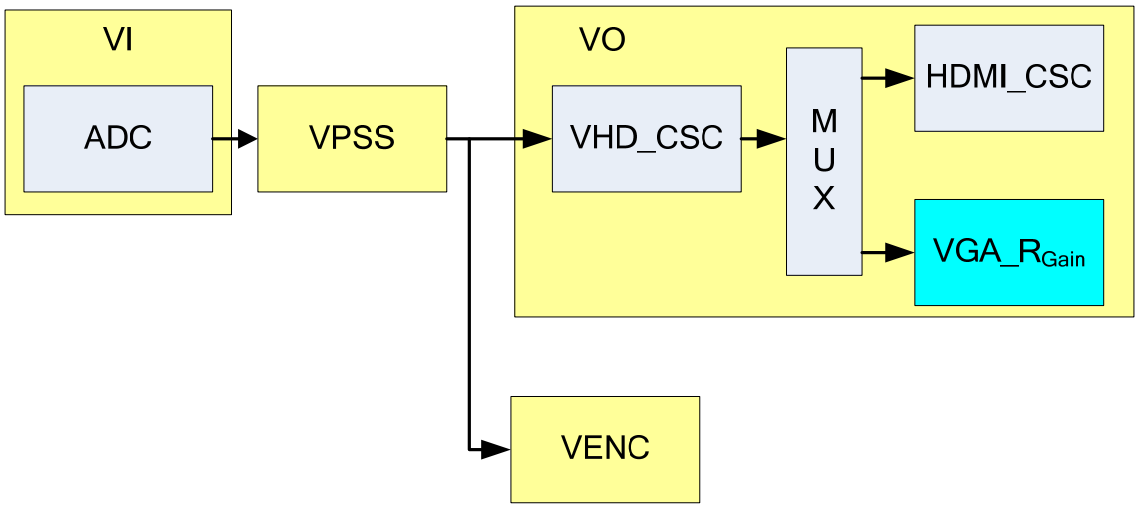
u32Contrast: contrast

u32Hue: hue

u32Saturation: saturation

Note that the Hi3520D, Hi3515A, or Hi3515C does not support the **u32Gain** parameter. Figure 1-1 shows the video processing workflow. The **VHD_CSC** parameter is adjusted when HI_MPI_VO_SetDevCSC() is called; therefore, the HDMI channel is not affected, and you need to adjust the HDMI effect by calling HI_MPI_HDMI_SetCsc().

Figure 1-1 Video processing workflow



If you want to change the default parameter values of the analog-to-digital converter (ADC) of the video input unit (VIU), you are advised to change the values based on the VENC picture effect because the VENC picture effect is closely related to ADC parameters. See Figure 1-2.

Figure 1-2 VGA test standard defined by Tektronix

Color Bars

Line = 145 Average = 1

Color Bars	G	B	R
White	705.904	705.864	696.541
Yellow	701.935	-3.77	693.01
Cyan	701.532	705.356	-3.405
Green	701.561	-3.488	-0.261
Magenta	-4.5	705.698	697.161
Red	0.473	-3.303	693.509
Blue	0.773	705.767	-3.369
Black	0.555	-2.975	-0.034

- If your VGA circuits are different from the reference VGA circuits delivered by HiSilicon, you can separately adjust the VGA effect by calling HI_MPI_VO_SetVgaParam for the Hi3531, Hi3532, Hi3521, or Hi3520A or by calling HI_MPI_VO_SetDevCSC() for the Hi3515A or Hi3520D. Note that HI_MPI_VO_SetVgaParam does not apply to the Hi3515A or Hi3520D.

  You can also change your VGA circuits based on reference VGA circuits delivered by HiSilicon. This section uses the Hi3521 circuits as an example. For the circuits of other chips, see the board design schematic diagrams. Note the following when you change VGA circuits:

  - Set R423 to 51 kΩ and R638 to 1.1 kΩ, and ensure that the resistor precision is 1%. See Figure 1-3.
  - If no video buffer (Z1 in Figure 1-4), keep the 75 Ω resistors R366, R363, and R367 not connected. See Figure 1-4.

Figure 1-3 R423 and R638 in the VGA circuit

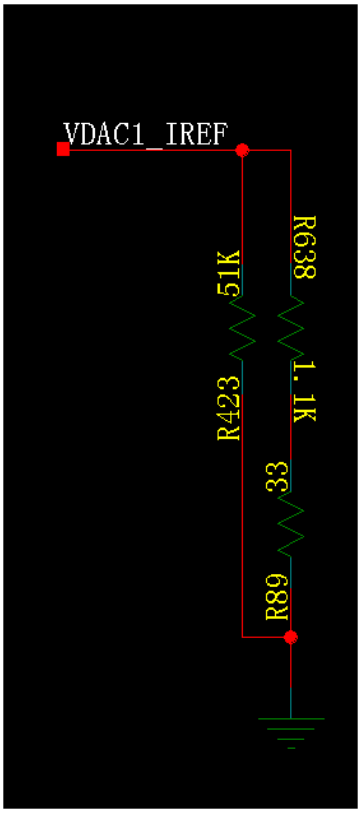
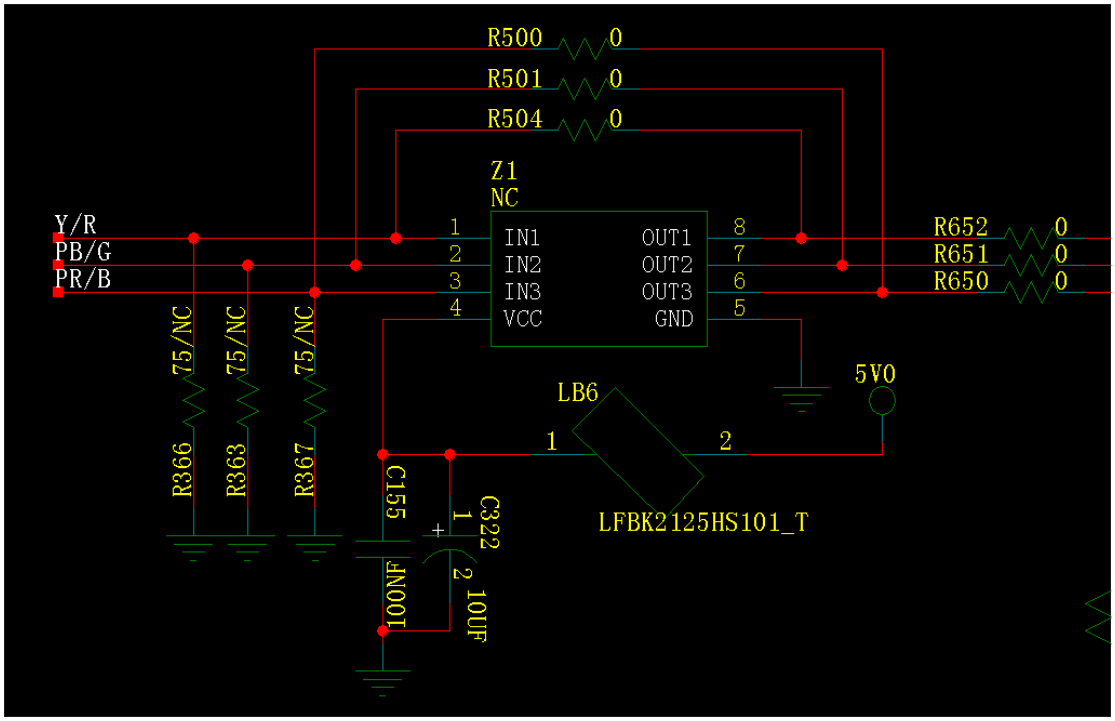


Figure 1-4 VGA output circuit



1.4 How Do I Adjust the Denoising Effect of the VPSS?

[Question]

How do I adjust the denoising effect of the VPSS for the Hi3520D, Hi3515A, or Hi3515C?

[Analysis]

The **TfStr** and **SfStr** parameters impose the same impacts on the four back-end physical channels of the VPSS, but the **Motion** parameter is valid only for the encoding channel.

[Solution]You are advised to set the **SfStr** and **TfStr** parameters at a ratio ranging from 2:1 to 4:1 for previewing channels. When the denoising effect of previewing channels is optimal, adjust the denoising effect of encoding channels.

- Adjust the **Motion** parameter for encoding channels. A larger **Motion** parameter value indicates higher denoising strength. However, if the **Motion** parameter value is too large, smearing occurs.