HiMPP V3.0

FAQs

Issue **09**

Date **2018-05-20**

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

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
Hi3536	V100
Hi3521A	V100
Hi3520D	V300
Hi3531A	V100
Hi3531D	V100
Hi3521D	V100
Hi3536C	V100
Hi3536D	V100
Hi3520D	V400

Intended Audience

This document is intended for:

- Technical support engineers
- 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 10 (2018-05-20)

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

Section 3.6.2 is modified.

Section 3.11 is added.

Issue 09 (2018-01-17)

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

Section 3.10 is added.

Issue 08 (2017-11-30)

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

Section 1.4.1 is added.

Issue 07 (2017-11-20)

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

Section 3.9 is added.

Issue 06 (2017-08-10)

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

The description of the Hi3536D V100 is added.

Section 1.9 is added.

Issue 05 (2017-07-26)

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

Section 3.8 is updated.

Issue 04 (2017-04-10)

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

The description of the Hi3536C V100 is added.

Section 3.7 is modified.

Section 3.8 is added.

Issue 03 (2017-01-06)

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

Chapter 3 Others

Section 3.5, section 3.6, and section 3.7 are added.

Issue 02 (2016-06-15)

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

Chapter 3 Others

Section 3.4 is added.

Issue 01 (2015-12-15)

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

Chapter 1 System Control

Section 1.6 is modified.

Section 1.8 is added.

Chapter 2 Audio

Sections 2.4 and section 2.5 are added.

Chapter 3 Others

Section 3.2 and section 3.3 are added.

Issue 00B02 (2015-08-20)

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

Chapter 1 System Control

Section 1.3, section1.4, and section 1.7 are added.

Chapter 2 Audio

Section 2.3 is added.

Chapter 3 Others

This chapter is added.

Issue 00B01 (2015-06-09)

This issue is the first draft release.

Contents

Figures

Tables

1 System Control

1.1 Log Information

1.1.1 How Do I View HiMPP Logs?

[Symptom]

How do I view HiMPP logs and change the log level?

[Cause 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/logmpp** 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/logmpp** 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/logmpp** command.
- To obtain logs, run the **cat /dev/logmpp** 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/logmpp**, run **open** and **read** commands.
- The preceding commands can be used for debugging all modules when a single CPU is used, and for debugging modules of the slave ARM when dual CPUs are used.
- When dual CPUs are used, the log level and log information of each module of the slave ARM can be controlled over the serial ports of the slave ARM or by the master ARM.

- To view the log level of each module of the master ARM when dual CPUs are used, run **cat /proc/umap/mstlogmpp**. To change the log level of a module, run **echo "venc=4" >/proc/umap/mstlogmpp**. To change the log levels of all the modules, run **echo "all=4" >/proc/umap/mstlogmpp**.
- To obtain logs on the master ARM, run **cat /dev/mstlogmpp**.

1.2 Memory Usage

1.2.1 How Do I Adjust the Memories Occupied by Media Services?

[Symptom]

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.

[Cause Analysis]

The HiSilicon SDK allows you to adjust the allocated memories when the memories are insufficient. This section briefly describes the measures to minimize memory usage. For details, see related documents.

[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**.
- Adjust the memories occupied by SDK services.
  - Entire system

    Ensure that the size of a D1-series (D1, 2CIF, and CIF) picture is an integral multiple of the sizes of the other D1-series pictures. For example, the size of a D1 picture is 704 x 576, the size of a 2CIF picture is 352 x 576, and the size of a CIF picture is 352 x 288. If the size of a 2CIF picture is 352 x 576 and the size of a CIF picture 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 and the size of the picture encoded by a VENC channel is 704 x 576, the picture sizes do not meet requirements.
  - Minimum buffer size for each module

    For details, see the *HiMPP V3.0 Media Processing Software Development Reference*.
  - Just enough public video buffers (VBs)

    See HI_MPI_VB_SetConf.

    For details, see chapter 2 "System Control" in the *HiMPP V3.0 Media Processing Software Development Reference*.

    In the proc information about the VB, if **IsComm** is **1**, the VB pool is a public VB pool. If **MinFree** of the public VB pool is **0** and there is no displayed information indicating that a module cannot obtain the VB in the logmpp, the public VBs are just enough.
  - Video processing subsystem (VPSS)

    For details, see chapter 5 "VPSS" in the *HiMPP V3.0 Media Processing Software Development Reference*.

Measure	MPI	Benefit	Impact	Note
Disable backup frames.	HI_MPI_VPSS_EnableBackupFrame, HI_MPI_VPSS_DisableBackupFrame	Each VPSS group spares one frame buffer.	If the picture is switched when the video output unit (VOU) pauses, the background color of the device is displayed.	None
Disable the noise reduction (NR) and deinterlacing (DEI) functions simultaneously.	HI_MPI_VPSS_CreateGrp	Each VPSS group spares the buffers for two frames (reference frame and reconstruction frame).	The picture quality is affected.	If any of the NR and DEI functions is enabled, the buffer for the reference frame and reconstruction frame will be allocated.

– Video encoding (VENC)

For details, see chapter 6 "VENC" in the *HiMPP V3.0 Media Processing Software Development Reference*.

Measure	MPI/Parameter	Benefit	Impact	Note
Dynamically switch the encoding resolution.	HI_MPI_VENC_GetChnAttr, HI_MPI_VENC_SetChnAttr	The VENC channel is not destroyed when the encoding resolution is switched, which reduces memory fragments.	None	After the encoding resolution is switched, all parameters are restored to default values.
Configure the allocation mode of the buffers for storing encoded frames.	H264eVBSource	The memory usage is reduced.	For details, see section 6.2.10 in chapter 6 "VENC" in the *HiMPP V3.0 Media Processing Software Development Reference*.	For details, see section 6.2.10 in chapter 6 "VENC" in the *HiMPP V3.0 Media Processing Software Development Reference*.

Measure	MPI/Parameter	Benefit	Impact	Note
Enable the reference frame to share the memory for storing the luminance data with the reconstruction frame by setting the corresponding parameter.	H264eRcnEqualRef	The memory usage is reduced.	For details, see section 6.2.11 in chapter 6 "VENC" in the *HiMPP V3.0 Media Processing Software Development Reference*.	For details, see section 6.2.11 in chapter 6 "VENC" in the *HiMPP V3.0 Media Processing Software Development Reference*.

– Video decoding (VDEC)

For details, see chapter 10 "VDEC" in the *HiMPP V3.0 Media Processing Software Development Reference*.

Measure	MPI/Parameter	Benefit	Impact	Note
Set the number of display frames to **0**.	HI_MPI_VDEC_GetProtocolParam, HI_MPI_VDEC_SetProtocolParam	The buffer for storing display frames does not need to be allocated for the VDEC channel.	The VO display smoothness is affected.	• The VPSS backup frames must be disabled. • The number of VDEC display frames cannot be set to 0 if the decoded picture is transmitted to a standard definition (SD) VO device. • The module VB pool, private VB pool, and user VB pool differ in buffer allocation mode.
Set the number of reference frames in the VDEC channel that decodes only I frames to **0**.	HI_MPI_VDEC_CreateChn	The buffer for storing reference frames does not need to be allocated.	None	Set the channel decoding mode to I mode (only I frames are decoded); otherwise, an error occurs in the logmpp.
Disable the B frame decoding	HI_MPI_VDEC_CreateChn	The buffer for outputting the Pmv	None	The module VB pool, private VB pool, and user VB pool differ in

Measure	MPI/Parameter	Benefit	Impact	Note
function in the VDEC channels that do not need to decode B frames.		information does not need to be allocated for the VDEC channel.		buffer allocation mode.
Allocate the stream buffers in reduction mode.	MiniBufMode	The stream buffer size can be reduced.	You need to discard the jumbo frame whose size exceeds the stream buffer size.	This measure is valid only when streams are decoded by frame.

– VO

For details, see chapter 4 "VO" in the *HiMPP V3.0 Media Processing Software Development Reference*.

Measure	MPI	Benefit	Impact	Note
Set the display queue length to the minimum value **3**.	HI_MPI_VO_GetDispBufLen, HI_MPI_VO_SetDispBufLen	The Hi3536 high definition (HD) device spares one frame buffer.	The VO display smoothness is affected.	None
Set the size of the canvas at the PIP layer for the HD device to the size of the displayed picture.	HI_MPI_VO_GetVideoLayerAttr, HI_MPI_VO_SetVideoLayerAttr, HI_MPI_VO_GetChnDispPos, HI_MPI_VO_SetChnDispPos	The memory performance is improved by setting **stImageSize** to the size of the displayed picture.	None	The automatic zoom-in function of the VOU is not supported at the PIP layer.
Use the automatic zoom-in function of the VOU.	HI_MPI_VO_GetVideoLayerAttr, HI_MPI_VO_SetVideoLayerAttr	If **stImageSize** is less than **stDispRect**, the HD VO device at the video layer can use the automatic zoom-in function, which reduces memory usage and bandwidth.	None	The SD device does not support scaling.

- Video graphics subsystem (VGS)

  For details, see chapter 13 "VGS" in the *HiMPP V3.0 Media Processing Software Development Reference*.

Measure	Parameter	Benefit	Impact	Note
Set the maximum number of jobs supported by the VGS.	max_vgs_job	The size of the memory occupied by a job is 196 bytes.	The VGS performance is restricted if the number of jobs is too small.	None
Set the maximum number of tasks supported by the VGS.	max_vgs_task	The size of the memory occupied by a task is 1740 bytes.	The VGS performance is restricted if the number of tasks is too small.	None
Set the maximum number of nodes supported by the VGS.	max_vgs_node	The size of the memory occupied by a node is 16596 bytes.	The VGS performance is restricted if the number of nodes is too small.	None

- HiFB

  For details, see the *HiFB Development Guide*.

Measure	MPI/Parameter	Benefit	Impact	Note
Set an appropriate size for the physical display buffer at the graphics layer.	Video	Setting an appropriate size for the physical display buffer at the graphics layer according to the actual resolution avoids memory waste.	None	If a VO device is not used, the physical display buffer at the corresponding graphics layer does not need to be allocated.

1.3 How Does the Master ARM Access the MMZ Memory of the Slave ARM in the Dual-CPU Version?

In the dual-CPU version, the MMZ memory obtained by calling HI_MPI_SYS_MmzAlloc or HI_MPI_SYS_MmzAlloc_Cached is the MMZ memory at the master ARM end. To use the MMZ memory of the slave ARM, you are advised to perform the following steps:

Step 1 Call HI_MPI_VB_CreatePool to create a VB pool in the MMZ memory of the slave ARM.

Step 2 Call HI_MPI_VB_MmapPool to map the physical address of the VB pool if the virtual address corresponding to the physical address of the VB pool is required.

Step 3 Call HI_MPI_VB_GetBlock to obtain a piece of memory from the VB pool.

Step 4 Call HI_MPI_VB_ReleaseBlock to release the memory back to the VB pool after the memory is used.

Step 5 Call HI_MPI_VB_DestroyPool to destroy the VB pool.

----End

1.4 Performance

1.4.1 Function and Impact of USB Priority Adjustment

In the XVR platform, if the mouse pointer automatically drifts, you can improve the USB module priority to solve this problem. To adjust the USB module priority, do as follows:

- For the Hi3536, Hi3531A, and Hi3521A, modify the USB module priority in the **syscrtl.sh** script in the chip solution release package.
- For the Hi3531D V100, Hi3521D V100, Hi3520D V400, Hi3536C V100, and Hi3536D V100, modify the USB module priority in the **arch\arm\boot\dts\Hi35XX-demb.dts** file in the **kernel** directory.

1.4.2 Function and Impact of CPU Priority Adjustment

In the dual-CPU system, the QoS mapping value of the A7 CPU can be changed to adjust the A7 performance. For details, see the description of the MISC_CTRL79 register in the *Hi3536 H.265 Decoder Processor Data Sheet* and the default script **syscrtl.sh** in the chip solution release package.

NOTE

The QoS mapping values of various IPs in the **syscrtl.sh** script of the chip solution release package are a group of configured values that ensures balanced overall performance. You are advised not to change the default values except under special circumstances. If the QoS mapping value of a single IP is adjusted separately, the performance of other IPs may be affected.

1.5 Miniaturization

1.5.1 How Do I Reduce the Application Size When the Static Libraries Are Used?

[Symptom]

The application uses only a small part of the functions in the **libmpi.a** library. However, the application needs to link to the associated library files such as **vqev2**. As a result, the application size is too large.

[Cause Analysis]

The application needs to link to all the functions defined in the MPI libraries by default when it links to the MPI libraries. Therefore, the application needs to use other libraries associated with the MPI libraries.

[Solution]

When the libraries of the HiMPP are generated, add the **-ffunction-sections** compilation option to **Makefile.param**. When the application links to the MPI libraries during compilation, add the **-Wl,-gc-sections** compilation option to **Makefile**. This deletes the functions that are not used and reduces the application size significantly.

1.6 Virtual Serial Port

1.6.1 How Do I Use the Virtual Serial Port?

[Symptom]

The VTTY debugging tool is provided in the SDK.

[Cause Analysis]

VTTY acts as a virtual serial port that can replace the actual serial port and be used for debugging.

[Solution]

- The source code required by VTTY is stored in **component/vtty** of the SDK. After the kernel path in **Makefile** is modified, you can run the **make** command to perform compilation. After the compilation is complete, **vcom_tty.ko** and **vshell** are generated. **vcom_tty.ko** runs on the slave ARM and is stored in the file system of the slave ARM by default; **vshell** runs on the master ARM.
- Add **vtty0::respawn:-/bin/login root** to the **inittab** configuration file in the file system of the slave ARM under **rootfs-SLV_FULL_REL/etc/**. You can comment out echo "vtty0::respawn:-/bin/login root" >> $SLAVE_FS_DIR/etc/inittab by referring to the **mpp_mkfs**.sh script.
- Comment out **#insmod vcom_tty.ko init_id=238** and **./dccs_main 1>/dev/vtty0 2>/dev/vtty0 &**, and delete **./dccs_main&** in the **run_slave** script (in the file system of the slave ARM) under **rootfs-SLV_FULL_REL/etc/init.d/S36autostart**. Note that **init_id** indicates the PID value of the system init thread.

- Run **vshell** on the Telnet terminal or the terminal connected to the serial port of the master ARM.

1.7 How Do I Enable and Disable the System Performance Statistics Function?

To reduce system power consumption, the system performance statistics function is disabled by default. After this function is enabled, information such as the interrupt handling time and interrupt reporting interval can be viewed, which facilitates user debugging and fault positioning. You can enable and disable this function by using either of the following methods:

- On the serial port of the master ARM or the Telnet terminal, run **echo "657676" > /proc/sys/all/msg/proc_all** to enable the performance statistics function of each module in the master ARM system, and run **echo "1" > /proc/sys/all/msg/proc_all** (the value in double-quotes can be any value except 657676) to disable the performance statistics function of each module in the master ARM system.
- On the serial port of the slave ARM or the virtual serial port, run **echo "657676" > /proc/sys/all/msg/proc_all** to enable the performance statistics function of each module in the slave ARM system, and run **echo "1" > /proc/sys/all/msg/proc_all** (the value in double-quotes can be any value except 657676) to disable the performance statistics function of each module in the slave ARM system.

1.8 Precautions During Video Cascade Configuration

- During video cascade, the VOU outputs the standard timing signal and the VIU parses the timing. In this way, data transfer is implemented.
- When the VIU parses the BT.1120 standard timing, 0xff 00 00 is used as the data of the synchronization header signal. When the VOU generates the timing signal, the status information is written in the blanking region, which is used by software to manage the cascade status. Note that the status information of the synchronization header value should be avoided in the blanking region during data transfer. Otherwise, errors may occur when the VIU parses the timing, and the transfer may fail.
- When calling HI_MPI_VO_SetCascadePattern, avoid setting **u32Pattern** to **0x7f**. Otherwise, the 0xff 00 00 synchronization header may occur.

1.9 Precautions To Be Taken for the Asymmetric Dual-channel DDR

[Symptom]

When the customer uses 1.5 GB or 3 GB DDR memory for the Hi3536 or Hi3531A chip based on the product form, there is 0.5 GB or 1 GB DDR memory in a single channel, which degrades the system performance.

[Analysis]

A DDR single channel has lower performance than the dual channels, which causes degradation of the system performance.

[Solution]

To ensure the normal running of media services, it is advisable to allocate the entire DDR memory of a single channel to the OS while media services use the dual-channel DDR memory. Note that the specific performance needs to be evaluated according to the actual scenario.

[Precaution]

When only 3 GB memory is used for a customer board attached with 4 GB memory, the configurations should be performed according to the 4 GB DDR memory table. The recommended configuration for the precaution can be ignored.

2 Audio

2.1 How Do I Play the Audio Streams Encoded by HiSilicon on the PC?

2.1.1 How Do I Play G711/G726/ADPCM Audio Streams Encoded by HiSilicon on the PC?

[Symptom]

The G711/G726/ADPCM audio streams encoded by HiSilicon cannot be played directly by using software on the PC.

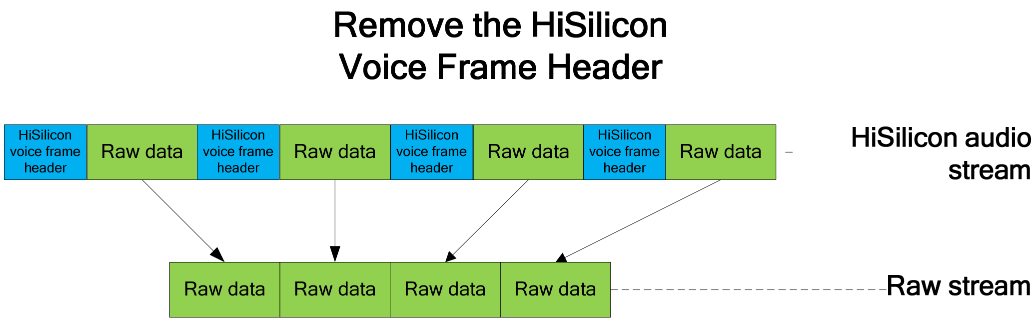
[Cause Analysis]

A HiSilicon voice frame header is added at the beginning of each frame in the audio streams encoded by HiSilicon. For details, see section 9.2.2.3 in the *HiMPP V3.0 Media Processing Software Development Reference*. The HiSilicon voice frame header cannot be identified by software on the PC.

[Solution]

Remove the HiSilicon voice frame header at the beginning of each frame, add the WAV header to the frames in the raw stream, and play the streams using software on the PC. Figure 2-1 shows how to remove the HiSilicon voice frame header.

Figure 2-1 Remove the HiSilicon voice frame header



The reference code for removing the HiSilicon voice frame header is as follows:

```
int HisiVoiceGetRawStream(short *Hisivoicedata, short *outdata, int
hisisamplelen)
{
    int len = 0, outlen = 0;
    short *copyHisidata, *copyoutdata;
    int copysamplelen = 0;

    copysamplelen = hisisamplelen;
    copyHisidata = Hisivoicedata;
    copyoutdata = outdata;

    while(copysamplelen > 2)
    {
        len = copyHisidata[1]&0x00ff;
        copysamplelen -= 2;
        copyHisidata += 2;
        if(copysamplelen < len)
        {
            break;
        }
        memcpy(copyoutdata, copyHisidata, len * sizeof(short));
        copyoutdata += len;
        copyHisidata += len;
        copysamplelen -= len;
        outlen += len;
    }

    return outlen;
}
```

NOTE

- The audio streams in ADPCM_DVI4 or ADPCM_ORG_DVI4 format are used for network transfer over the Real-time Transport Protocol (RTP) and cannot be played by the client programs on the PC. For details, see the RFC35551 standard.
- The method of adding the WAV header is not provided in this document. You can add the WAV header by following the WAV header standard. For details, see the reference links https://msdn.microsoft.com/en-us/library/dd390970(v=vs.85).aspx and http://www.moon-soft.com/program/FORMAT/windows/wavec.htm.

2.2 How Do I Play Standard Audio Streams on HiSilicon Chips?

2.2.1 How Do I Play Standard G711/G726/ADPCM Audio Streams on HiSilicon Chips?

[Symptom]

The standard G711/G726/ADPCM audio streams cannot be played directly on HiSilicon chips.

[Cause Analysis]

To ensure that the previous-generation chips are compatible, the audio streams can be played on HiSilicon chips only after the HiSilicon voice frame header is added at the beginning of each frame in the raw audio streams.

[Solution]

To play G711/G726/ADPCM audio streams on HiSilicon chips, obtain the raw stream data, add the HiSilicon voice frame header at the beginning of each frame based on the frame data length **PerSampleLen**.

Step 1 Obtain the raw stream data. Remove the WAV header if the WAV header is added to the frame.

Step 2 Obtain the data length of each frame (**PersampleLen**, a short number).

Table 2-1 Data length of each frame

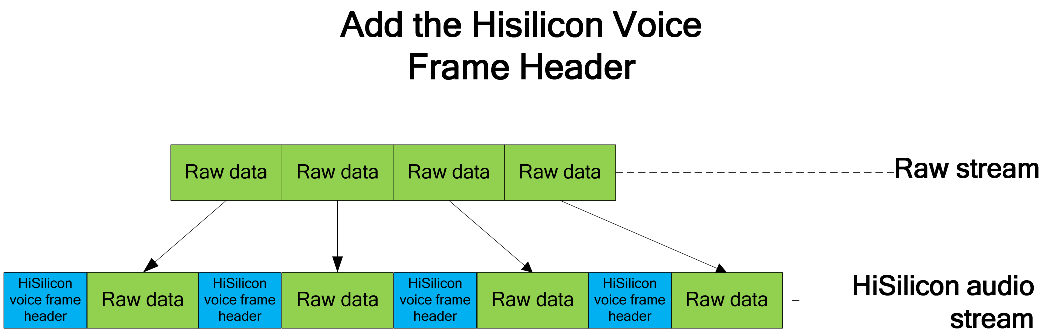
Encoding Format	Data length of each frame	Remarks
G711	*N* x 40	*N* is a positive integer ranging from 1 to 5.
G726 (16 kbit/s)	*N* x 10	*N* is a positive integer ranging from 1 to 5.
G726 (24 kbit/s)	*N* x 15	*N* is a positive integer ranging from 1 to 5.
G726 (32 kbit/s)	*N* x 20	*N* is a positive integer ranging from 1 to 5.
G726 (40 kbit/s)	*N* x 25	*N* is a positive integer ranging from 1 to 5.
IMA ADPCM	Number of bytes in each block/2	The number of bytes in each block indicates the number of bytes in the encoded IMA ADPCM data of each block, corresponding to **nblockalign** (0x20–0x21, 2-byte) of the IMA ADPCM WAV header.

NOTE

- Of all the ADPCM formats, only the IMA ADPCM format is supported. The number of bytes in each sampling point (**wbitspersample**) must be 4.
- If the WAV header is added to the frames in ADPCM streams, the number of bytes in each block can be obtained from the WAV header. For the raw ADPCM streams, the number of bytes in each block must be obtained from the provider of the streams.
- Only the mono-channel encoding format is supported.

Step 3 Add the HiSilicon voice frame header, as shown in Figure 2-2.

Figure 2-2 Add the HiSilicon voice frame header



The reference code for adding the HiSilicon voice frame header is as follows:

```
int HisiVoiceAddHisiHeader(short *inputdata, short *Hisivoicedata, int
PersampleLen,int inputsamplelen)
{
    int len = 0, outlen = 0;
    short HisiHeader[2];
    short *copyHisidata, *copyinputdata;
    int copysamplelen = 0;
    HisiHeader[0] = (short)(0x001<<8) & (0x0300);
    HisiHeader[1] = PersampleLen & 0x00ff;
    copysamplelen = inputsamplelen;
    copyHisidata = Hisivoicedata;
    copyinputdata = inputdata;
    while(copysamplelen >= PersampleLen)
    {
        memcpy(copyHisidata, HisiHeader, 2 * sizeof(short));
        outlen += 2;
        copyHisidata += 2;
        memcpy(copyHisidata, copyinputdata, PersampleLen * sizeof(short));
        copyinputdata += PersampleLen;
        copyHisidata += PersampleLen;
        copysamplelen -= PersampleLen;
        outlen += PersampleLen;
    }
    return outlen;
}
```

----End

2.3 What Do I Do If High-Frequency Information Is Lost After VQE Is Enabled?

2.3.1 What Do I Do If High-Frequency Information Is Lost After VQE Is Enabled?

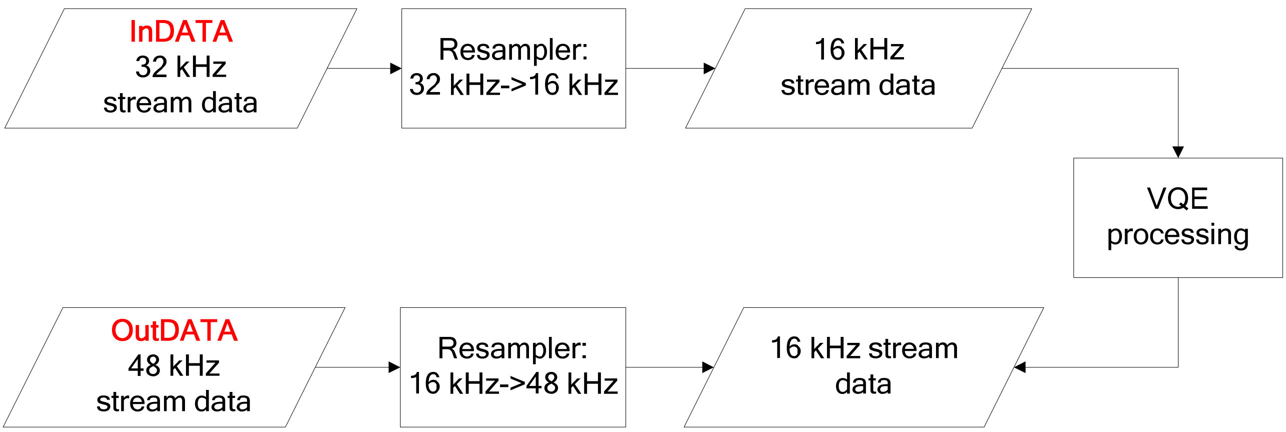
[Symptom]

When the AI sampling rate (**AISampleRate**) is 32 kHz, voice quality enhancement (VQE) working sampling rate (**VQEWorkSampleRate**) is 16 kHz, output sampling rate (**ResOutSampleRate**) is 48 kHz, and the VQE and resampling functions are enabled, the 8 kHz or higher-frequency information is lost according to the analysis result of the output sequence.

[Cause Analysis]

In actual applications, the HiVQE supports only the 8 kHz and 16 kHz working sampling rates. To meet customer requirements, the resampling layer is encapsulated in the HiVQE to support any standard sampling rate from 8 kHz to 48 kHz. When **AISampleRate**, **VQEWorkSampleRate**, and **ResOutSampleRate** are set to **32 kHz**, **16 kHz**, and **48 kHz** respectively, the resampling layer resamples data from 32 kHz to 16 kHz, and then resamples data from 16 kHz to 48 kHz for output after VQE processing. Figure 2-3 shows the VQE processing flow.

Figure 2-3 VQE processing flow



When data is resampled from 32 kHz to 16 kHz, 8 kHz or higher-frequency information is lost.

In the current application scenario, the frequency band information of the output sequence is as follows:

- When VQE and resampling are disabled, the information within the frequency band of 0 to (**AISampleRate**/2) is output. For example, if the AI sampling rate (**AISampleRate**) is 48 kHz, the frequency band of the output information is 0 kHz to 24 kHz.
- When VQE is disabled and resampling is enabled, the information within the frequency band of 0 to min(**AISampleRate**, **ResOutSampleRate**)/2 is output. For example, if the AI sampling rate (**AISampleRate**) is 16 kHz and the output sampling rate

(**ResOutSampleRate**) is 32 kHz, min(**AISampleRate**, **ResOutSampleRate**) is 16 kHz. Therefore, the frequency band of the output information is 0 kHz to 8 kHz.

- When VQE is enabled and resampling is disabled, the information within the frequency band of 0 to min(**AISampleRate**, **VQEWorkSampleRate**)/2 is output. For example, if the AI sampling rate (**AISampleRate**) is 32 kHz and the VQE working sampling rate (**VQEWorkSampleRate**) is 16 kHz, min(**AISampleRate**, **VQEWorkSampleRate**) is 16 kHz. Therefore, the frequency band of the output information is 0 kHz to 8 kHz.
- When VQE and resampling are enabled, the information within the frequency band of 0 to min(**AISampleRate**, **VQEWorkSampleRate**, **ResOutSampleRate**)/2 is output. For example, if the AI sampling rate (**AISampleRate**) is 32 kHz, the VQE working sampling rate (**VQEWorkSampleRate**) is 16 kHz, and the output sampling rate (**ResOutSampleRate**) is 48 kHz, min(**AISampleRate**, **VQEWorkSampleRate**, **ResOutSampleRate**) is 16 kHz. Therefore, the frequency band of the output information is 0 kHz to 8 kHz.

NOTE

- After the sampling rate is configured, the maximum frequency of the output information is half of the sampling rate.
- The standard sampling rates from 8 kHz to 48 kHz are supported, including 8 kHz, 11.025 kHz, 12 kHz, 16 kHz, 22.05 kHz, 32 kHz, 44.1 kHz, and 48 kHz.
- The processing flow of the audio output unit (AOU) is similar to that of the audio input unit (AIU).

2.4 What Do I Do If Pop Sound Occurs When G726 Streams Are Played?

[Symptom]

If the encoding channel is repeatedly destroyed and created during the process of generating G726 streams, the pop sound occurs when the obtained streams are decoded and played.

[Cause Analysis]

During the encoding/decoding using the G726 protocol, the history information of the previous frame needs to be used for predicting encoding/decoding. The history information is continuously updated during the encoding/decoding of each frame. If the encoder is suddenly restarted, all the history information will be reset. As the history information after reset is used during re-encoding, the data before reset cannot be joined together with the data after reset, and exceptions occur. During decoding, the pop sound occurs due to the abnormal encoding data at the position where the encoder/decoder is reset. The pop sound occurs when non-HiSilicon decoder is used for decoding. If the encoding protocol that needs to rely on the information of the previous frame is used, it is recommended that all the audio-related channels be reset when the encoder/decoder is reset.

2.5 What Do I Do If the Amplitude-Frequency Response Is Abnormal During the Audio Embedded CODEC Output (AO Output)?

[Symptom]

When the amplitude-frequency response of the audio CODEC (DAC) output (AO output) is tested, the amplitude-frequency response of the 2 kHz and higher frequencies attenuates significantly.

[Cause Analysis]

This issue is caused because dacl_deemph bit[22:21] and dacr_deemph bit[20:19] of the audio CODEC control register are configured to enable the de-emphasis function. (Theses bits in the audio CODEC control register of some chips are not opened.) De-emphasis is relative to pre-emphasis, and is modification on pre-emphasis. If the audio input signal of the AO channel is after pre-emphasis, enabling the de-emphasis function restores the amplitude-frequency response to normal. If the audio input signal of the AO channel is not after pre-emphasis but the dacl_deemph and dacr_deemph register fields enable the de-emphasis function, the amplitude-frequency response will be affected. This test is conducted when the HIVQE function is disabled. During the test, pre-emphasis is not performed on the data output to the AO channel, but dacl_deemph bit[22:21] and dacr_deemph bit[20:19] of the audio CODEC control register are not disabled (neither of them is 00). As a result, this issue occurs.

[Solution]

For the AI channels, the pre-emphasis function is not enabled in the audio CODEC control register. Therefore, the dacl_deemph and dacr_deemph fields of the audio CODEC control register need to be disabled and set to 00. Note that the pre-emphasis function should be used together with the de-emphasis function.

[Notes]

Only the Hi3536 supports the embedded CODEC.

3 Others

3.1 Dynamic Library

3.1.1 What Do I Do If the Dynamic Libraries Cannot Be Used When the Application Is Statically Compiled?

[Symptom]

The file systems and executable programs of customer A are statically compiled. As a result, the dynamic libraries in the SDK cannot be used.

[Cause Analysis]

The current arm-linux-gcc version supports the static compilation, dynamic compilation, and semi-static compilation.

- In static compilation mode (compilation options: **-static**, **-pthread**, **-lrt**, and **-ldl**), the **libc**, **libpthread**, **librt**, and **libdl** libraries are all compiled to the executable program. The static compilation does not depend on any system dynamic library and is implemented independently. However, the dynamic libraries cannot be used in this mode.
- In dynamic compilation mode (common compilation), the system dynamic libraries under **/lib** are linked. Therefore, the compiled program depends on the system dynamic libraries. The advantage of the dynamic compilation is that the system dynamic libraries can be shared by multiple executable programs such as the BusyBox and Himount under **/bin**.
- In semi-static compilation mode (compilation options: **-static-libgcc**, **-static-libstdc++**, **-L**, **-pthread**, **-lrt**, and **-ldl**), the **libgcc** and **libstdc++** libraries are compiled to the executable program. Other system libraries still depend on the system dynamic libraries. In this mode, the dynamic libraries can be used, but the **libc**, **libpthread**, **librt**, and **libdl** files still need to be placed under the system directory.

[Solution]

Adopt the dynamic compilation and place the system files that the dynamic libraries depend on (including **ld-uClibc.so.0**, **libc.so.0**, **libpthread.so.0**, **librt.so.0**, and **libdl.so.0**) under **/lib**.

3.1.2 What Do I Do If a Redefinition Error Occurs When libupvqe.a and libdnvqe.a Are Used for Dynamic Compilation?

[Symptom]

A redefinition error occurred when customer B compiled the audio component libraries **libupvqe.a** and **libdnvqe.a** into a dynamic library. The compilation statement is as follows:

```
$(CC) -shared -o $@ -L. -Wl,--whole-archive libupvqe.a libdnvqe.a -Wl,--no-whole-archive
```

[Cause Analysis]

libupvqe.a and **libdnvqe.a** share some functional modules to implement code reuse and modularization, and save the file space when ELF files are generated after compilation.

The static libraries can be compiled into the dynamic library in any of the following ways:

- Directly use the **-l** compilation option. The compilation statement is as follows:

```
$(CC) -shared -o libshare.so -L. -lupvqe -ldnvqe
```

This is a link compilation method. The function symbols of the static libraries are not linked to the **libshare.so** library generated after compilation.

- Use the **-Wl,--whole-archive** compilation option. The compilation statement is as follows:

```
$(CC) -shared -o $@ -L. -Wl,--whole-archive libupvqe.a libdnvqe.a -Wl,--no-whole-archive
```

In this method, the function symbols of the static libraries are compiled to **libshare.so**. However, **libupvqe.a** and **libdnvqe.a** cannot have functions with the same name.

- Split the .a files into multiple .o files respectively, and then compile the .o files into the .so file. The compilation statement is as follows:

```
LIB_PATH = ./
EXTERN_OBJ_DIR = ./EXTERN_OBJ
LIBUPVQE_NAME = libupvqe.a
LIBDNVQE_NAME = libdnvqe.a
EXTERN_OBJ = $(EXTERN_OBJ_DIR)/*.o
all: pre_mk $(TARGET) pre_clr
pre_mk:
    @mkdir -p $(EXTERN_OBJ_DIR);
    @cp $(LIB_PATH)$(LIBUPVQE_NAME) $(EXTERN_OBJ_DIR);
    @cd $(EXTERN_OBJ_DIR); $(AR) -x $(LIBUPVQE_NAME);
    @cp $(LIB_PATH)$(LIBDNVQE_NAME) $(EXTERN_OBJ_DIR);
    @cd $(EXTERN_OBJ_DIR); $(AR) -x $(LIBDNVQE_NAME);

$(TARGET):
    #$(CC) -shared -o $@ -L. libupvqe.so libdnvqe.so
    $(CC) -shared -o $@ -L. $(EXTERN_OBJ)

pre_clr:
@rm -rf $(EXTERN_OBJ_DIR);
```

NOTE

In this method, **libupvqe.a** and **libdnvqe.a** are split into .o files respectively, and then the .o files are compiled into the .so file. The function symbols of the static libraries are compiled to the .so file, and no function name conflict occurs.

[Solution]

Customer B used the second compilation method. A redefinition error occurred because **libupvqe.a** and **libdnvqe.a** have functions with the same name. To solve this issue, customer B can use the first or third compilation method to generate the **libshare.so** file.

3.2 What Do I Do If the Encoding Blocking Artifact Is Obvious in Infrared Mode?

[Symptom]

When the IES strength of the 3DNR module is strong in infrared mode, the overall glitch effect of the picture is obvious. As the picture textures are few in this mode, the glitch effect causes the flat region to be not flat enough and increases the pressure of the encoder. In addition, the luminance variance between frames is large when the light changes. As a result, the pressure of the encoder is high when the light changes (for example, when the people walk or the lens is covered).

[Solution]

Set the IES of the 3DNR module to 0 in infrared mode (the default value is 4). In this way, the encoding efficiency in infrared mode is significantly improved. Besides, to strengthen the NR effect, set the TFR (HTFR) of the channel with strong NR strength to 56.

3.3 Performance Optimization in the 3840 x 480 DVR Front-End Capture Interlaced Scenario

To avoid clock switching, the DVR front-end analog-to-digital converter (ADC) up scales the picture with low resolution (such as 960H) by four times horizontally and sends the scaled picture (3840 x 480) to the VIU for processing. In this case, the VIU is under high pressure, which affects system performance. To prevent the competition for the bus from affecting the performance of modules (such as the VOU) that have high requirements on real time, it is recommended that 1/4 pixel skip be directly performed in the VIU, instead of performing secondary scaling by using the VPSS. In this way, the bandwidth usage can be lowered, and the VPSS performance is saved.

3.4 Precautions to Be Taken During the Cascading Between the Hi3536 and the Hi3531

Exceptions occur during the cascading of the Hi3536 and the Hi3531 because they differ in the PCIe cascading structure. In this case, modify the cascading structure used by the Hi3531 as PCIV_PIC_S. This structure is the same as that used by the Hi3536, and therefore the cascading is normal.

3.5 What Do I Do If A Black Screen Is Displayed on the HDMI Monitor When the Definition of the Hi3536 Startup Picture Is Inconsistent with That of a Normal Service?

NOTE

For details about related interfaces, see the *HiMPP V3.0 Media Processing Software Development Reference*.

For Hi3536, if the VO definition of the startup picture is inconsistent with that of the normal service, a black screen is displayed on the HDMI monitor during HDMI playback. In this case, you need to remove and insert the HDMI cable or restart the device to recover the normal display. The application program needs to disable and then enable the HDMI to solve the problem. Therefore, if you are confronted with this problem, you are advised to call the following interfaces to disable the HDMI before you call HI_MPI_VO_Disable to disable the VO.

[Example]

```
/*Disable HDMI*/
HI_MPI_HDMI_Init();
HI_MPI_HDMI_Open(HI_HDMI_ID_0);
HI_MPI_HDMI_Start(HI_HDMI_ID_0);

HI_MPI_HDMI_Stop(HI_HDMI_ID_0);
HI_MPI_HDMI_Close(HI_HDMI_ID_0);
HI_MPI_HDMI_DeInit();

/*Disable VO.*/
HI_MPI_VO_Disable();

/* After the problem is processed as planned, enable the VO or HDMI.*/
…
```

For details, see the SAMPLE_VO_Preview_HDMI example in **sample_vo.c**.

3.6 How Do I Set the VO User Time Sequence?

3.6.1 Configuration of the User Time Sequence Structure

NOTE

For details about related interfaces, see the *HiMPP V3.0 Media Processing Software Development Reference*.

When you call HI_MPI_VO_SetPubAttr to set the output public attributes, you can select VO_OUTPUT_USER as the interface time sequence, and then configure user time sequence structure as required.

```
typedef struct tagVO_SYNC_INFO_S
```

```
{
    HI_BOOL  bSynm;     /* sync mode(0:timing,as BT.656; 1:signal,as LCD)
*/
    HI_BOOL  bIop;      /* interlaced or progressive display(0:i; 1:p) */
    HI_U8    u8Intfb;   /* interlace bit width while output */
    HI_U16   u16Vact ;  /* vertical active area */
    HI_U16   u16Vbb;    /* vertical back blank porch */
    HI_U16   u16Vfb;    /* vertical front blank porch */
    HI_U16   u16Hact;   /* horizontal active area */
    HI_U16   u16Hbb;    /* horizontal back blank porch */
    HI_U16   u16Hfb;    /* horizontal front blank porch */
    HI_U16   u16Hmid;   /* bottom horizontal active area */
    HI_U16   u16Bvact;  /* bottom vertical active area */
    HI_U16   u16Bvbb;   /* bottom vertical back blank porch */
    HI_U16   u16Bvfb;   /* bottom vertical front blank porch */
    HI_U16   u16Hpw;    /* horizontal pulse width */
    HI_U16   u16Vpw;    /* vertical pulse width */
    HI_BOOL  bIdv;      /* inverse data valid of output */
    HI_BOOL  bIhs;      /* inverse horizontal synch signal */
    HI_BOOL  bIvs;      /* inverse vertical synch signal */
} VO_SYNC_INFO_S;
```

Table 3-1 describes the parameters.

Table 3-1 Parameter descriptions

Parameter	Description
bSynm	Synchronization mode. This parameter is unused and reserved currently.
bIop	0: interlaced mode 1: progressive mode
u8Intfb	Interface output bit width. This parameter is unused and reserved currently.
u16Vact	Vertical active area (in line). It indicates the top vertical active area in interlaced output mode.
u16Vbb	Vertical blanking back porch (in line). It indicates the top vertical blanking back porch in interlaced output mode.
u16Vfb	Vertical blanking front porch (in line). It indicates the top vertical blanking front porch in interlaced output mode.
u16Hact	Horizontal active area (in pixel)
u16Hbb	Horizontal blanking back porch (in pixel)
u16Hfb	Horizontal blanking front porch (in pixel)
u16Hmid	Bottom vertical sync active pixel

Parameter	Description
u16Bvact	Bottom vertical active area (in line). It is active in interlaced mode.
u16Bvbb	Bottom vertical blanking back porch (in line). It is active in interlaced mode.
u16Bvfb	Bottom vertical blanking front porch (in line). It is active in interlaced mode.
u16Hpw	Width of the horizontal synchronization signal (in pixel)
u16Vpw	Width of the vertical synchronization signal (in line)
bIdv	Polarity of the data valid signal. It cannot be adjusted currently (active high by default).
bIhs	Polarity of the horizontal active signal 0: active high 1: active low
bIvs	Polarity of the vertical active signal 0: active high 1: active low

You need to search for and refer to the specific document about the time sequence for the user time sequence configuration. Note that the unit of each value needs to be consistent with the required unit.

3.6.2 Configuration of the CRG

You need to calculate the PLL coefficient according to the clock frequency of the time sequence first. Requirements on coefficients may vary based on different chips, and you need to calculate the coefficient according to the requirements of the chip and time sequence.

CAUTION

Each chip has an independent PLL. You can refer to the chip data sheet and configure the corresponding register.

Step 1 See the methods of calculating PLL frequencies in section 3.2.4 "PLL Configuration" in *Hi35xx H.264 CODEC Processor Data Sheet* or *Hi35xx H.265 CODEC Processor Data Sheet*.

Table 3-2 Methods of calculating PLL frequencies (taking Hi3521A as an example)

PLL Pin	Formula	Remarks
FREF	PLL input reference clock	The input clock must be 24 MHz.
FOUTVCO	FREF x (fbdiv + frac/2^24)/refdiv	

PLL Pin	Formula	Remarks
FOUTPOSTDIV	FOUTVCO/(pstdiv1 x pstdiv2)	None
FOUT1ph0	FOUTVCO/(pstdiv1 x pstdiv2 x 2)	None
FOUT2	FOUTVCO/(pstdiv1 x pstdiv2 x 4)	None
FOUT3	FOUTVCO/(pstdiv1 x pstdiv2 x 6)	None
FOUT4	FOUTVCO/(pstdiv1 x pstdiv2 x 8)	None

NOTE

- fbdiv: integral frequency multiplier
- frac: decimal frequency multiplier
- refdiv: frequency divider of the reference clock
- pstdiv1: level-1 output frequency divider
- pstdiv2: level-2 output frequency divider

The following uses VPLL0 as an example. If VPLL0 outputs the 162 MHz FOUT1PH0 clock to the VDP module, the configured values of the register are calculated as follows:

- If postdiv2 is 5 and postdiv1 is 1, the value of FOUTVCO is 1620 MHz based on the following formula:
  FOUT1PH0 = FOUTVCO/(pstdiv1 x pstdiv2 x 2)
- If refdiv is 2, fbdiv and frac can be calculated based on the following formula:
  FOUTVCO = FREF x (fbdiv + frac/2^24)/refdiv = 24 x (fbdiv + frac/2^24)/2 = 1620 MHz
  That is, fbdiv = 135 and frac = 000000

Step 2 Configure the obtained values to the corresponding PLL registers.

See Table 3-2. The following is an example:

Calculate the values of **fbdiv** and **frac** according to **psotdiv2**, **postdiv1**, and **refdiv**. Configure the obtained values to the corresponding register of VPLL0 (by calling HI_MPI_SYS_SetReg).

See the register PERI_CRG_PLL2 in section 3.2.4 "PLL Configuration" in *Hi35xx H.264 CODEC Processor Data Sheet* or *Hi35xx H.265 CODEC Processor Data Sheet*, as shown in Table 3-3.

Table 3-3 PERI_CRG_PLL2 register

[30:28]	RW	vpll0_postdiv2	Level-2 output divider of the VPLL0
[27]	RO	reserved	Reserved
[26:24]	RW	vpll0_postdiv1	Level-1 output divider of the VPLL0
[23:0]	RW	vpll0_frac	Decimal divider of the VPLL0

See the register PERI_CRG_PLL3 in section 3.2.4 "PLL Configuration" in *Hi35xx H.264 CODEC Processor Data Sheet* or *Hi35xx H.265 CODEC Processor Data Sheet*, as shown in Table 3-4.

Table 3-4 PERI_CRG_PLL3 register

[17:12]	RW	vpll0_refdiv	Divider of the VPLL0 reference clock
[11:0]	RW	vpll0_fbdiv	Integral multiplier of the VPLL0

Step 3 Set the value of vo_hd0_hdmi_clk_div or vo_hd1_hdmi_clk_div.

The two registers need to be reset for the Hi3521D V100, Hi3531D V100, Hi3536C V100, and Hi3536D V100. For details about the register addresses, see the description of the CRG register in the *Hi35xx H.265 CODEC Processor Data Sheet*.

[9:5]	RW	vo_hd1_hdmi_clk_div	HDMI clock frequency divider (1–32) of the VO HD1 channel (N + 1)
[4:0]	RW	vo_hd0_hdmi_clk_div	HDMI clock frequency divider (1–32) of the VO HD0 channel (N + 1)

Before setting the register, obtain the value of the register and modify the bit to be reset. For example, if you need to set only the frequency division of VHD0, then modify the 0–4 bits. Other bits remain unchanged. Then, perform the setting again. Do not modify the bits that do not need to be reset. Otherwise, the VO module has no output.

For details about how to set value of the register, see the following table.

PIXEL_CLK Range	vo_hd0_hdmi_clk_div/vo_hd1_hdmi_clk_div
21.25 MHz–50 MHz	0xf
50 MHz–100 MHz	0x7
100 MHz–200 MHz	0x3
200 MHz–340 MHz	0x1
340 MHz–600 MHz	0x0

----End

3.6.3 Configuration of the VO Device Frame Rate

The default frame rate of the user timing is 60 fps. You need to set the VO device frame rate based on the actual frame rate. HI_MPI_VO_SetDevFrameRate needs to be called before HI_MPI_VO_Enable and after HI_MPI_VO_SetPubAttr to set the device frame rate of the user timing.

3.7 How Do I Solve the HDMI Compatibility Issue of the 4K Display?

[Symptom]

When the HDMI is connected to the 4K display (Samsung u28e590d), colors on the display are abnormal and the screen has a pink cast. The value of **AVI Infoframe** is set to **enable** by using HI_MPI_HDMI_SetAttr, but the value in the HDMI proc information is **disable**.

[Cause Analysis]

After the HiSilicon chip is started, one HPD voltage of this type of displays changes from 5 V to 0 V and then to 5 V during the HDMI initialization. Other displays do not have this changing process. In this case, the HDMI fails to configure the AVI information to the register.

[Solution]

After HI_MPI_HDMI_SetAttr is called, wait about 1s, and then call HI_MPI_HDMI_Start to start the HDMI.

3.8 How Do I Obtain the Slave CPU meminfo/media-mem/vmallocinfo on the Master CPU?

NOTE

This function is supported only by version Hi3536 V100R001C0xSPC061 and later.

[Symptom]

For Hi3536, the master CPU can obtain the device proc information in the **/proc/umap/** directory of the slave CPU by running **cat /proc/umap/***, but cannot obtain the **/proc/media-mem**, **/proc/meminfo**, and **/proc/vmallocinfo** of the slave CPU.

[Solution]

The master CPU can obtain the /proc/media-mem, /proc/meminfo, and /proc/vmallocinfo of the slave CPU in the version Hi3536 V100R001C0xSPC061 and later. To obtain the information, perform the following steps:

Step 1 Modify the **S36autostart** file in the **slave** directory.

Change **./dccs_main &** to **./dccs_main 1 &**.

Step 2 Make a slave file system again.

Run the **mpp_mkfs_glibc.sh** or **mpp_mkfs_uclibc.sh** script to make the slave CPU file system image.

Step 3 Obtains the information of the slave CPU.

When the system is running, enter **cat /proc/umap/dccs** to the master CPU. Then, the **media-mem**, **meminfo**, **vmallocinfo**, and **dccs** information of the slave CPU is returned.

----End

3.9 What Do I Do If an HDMI Hot Plug and Power Consumption Issue Occurs?

[Symptom]

When no external HDMI device is connected or when the hot plug is in low-level state, enabling the HDMI output will generate extra power consumption.

[Solution]

Normally, the registered callback function is recommended for handling the HDMI hot plug event. When the hot plug event is reported by the driver, call HI_MPI_HDMI_Start to enable the HDMI output; otherwise, keep it disabled it to reduce the power consumption.

The hot plug is always in low-level state for some non-standard display devices. In this case, if the HDMI output is required, you may call HI_MPI_HDMI_Start to forcibly enable the HDMI. When the HDMI output is not required, call HI_MPI_HDMI_Stop to disable the HDMI output.

3.10 What Do I Do If a Transparency or Color Issue of the OSD Occurs?

[Symptom]

- When the VPSS is overlaid with a region, the pixel format is ARGB444, and the transparency value 0xffff of the color is set to **100%**, but the image is still a little bit transparent.
- When the VGS is overlaid with a region or an OSD in white blocks, the output block color is light.

[Solution]

OSDs in formats ARGB1555, ARGB4444, and ARGB8888 are supported. The OSD is converted into 8-bit format for chip processing. However, because ARGB1555 and ARGB4444 use less than 8 bits to represent each sample, the logic shifts the bits to 8 bits. Due to mismatched precision, the image is transparent despite a transparency value of **100%**, or the color of the pure color block is light.

3.11 What Do I Do If Black Screen Issues Occur in Some Channels After the VI Module Is Enabled?

[Symptom]

In dual-edge mode, no picture is displayed in some VI channels after the service is started.

[Solution]

Do not modify the default clock timing of the driver. If the timing is incorrect, the chip logic may be abnormal. Before the AD clock is switched, ensure that the VI device is in reset state.