



Hi3520 视频级联

## 开发指南

文档版本 00B10

发布日期 2009-09-30

BOM编码 N/A

深圳市海思半导体有限公司为客户提供全方位的技术支持，用户可与就近的海思办事处联系，也可直接与公司总部联系。

## 深圳市海思半导体有限公司

地址：深圳市龙岗区坂田华为基地华为电气生产中心 邮编：518129

网址：<http://www.hisilicon.com>

客户服务电话：+86-755-28788858

客户服务传真：+86-755-28357515

客户服务邮箱：[support@hisilicon.com](mailto:support@hisilicon.com)

**版权所有 © 深圳市海思半导体有限公司 2009。保留一切权利。**

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

### 商标声明



**HISILICON**、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

### 注意

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。



## 目 录

前 言.....	1
1 概述.....	1-1
2 级联配置.....	2-1
2.1 注意事项.....	2-1
2.2 配置流程.....	2-1
2.2.1 主片配置流程.....	2-2
2.2.2 从片 1 配置流程.....	2-2
2.2.3 从片 2 配置流程.....	2-2
3 API 参考 .....	3-1
4 应用示例.....	4-1
4.1 主片代码示例 .....	4-1
4.2 从片 1 代码示例 .....	4-4
4.3 从片 2 代码示例 .....	4-7



## 插图目录

图 1-1 视频级联示意图.....	1-1
--------------------	-----



# 前言

## 概述

本文通过详细描述视频级联的配置流程及配置过程中的注意事项、在 API 参考的基础上给出应用示例，为实现 Hi3520 的视频级联功能提供参考。

## 产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3520 H.264 编解码处理器	V100

## 读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

## 约定

### 通用格式约定

格式	说明
宋体	正文采用宋体表示。
黑体	一级、二级、三级标题采用黑体。
楷体	警告、提示等内容一律用楷体，并且在内容前后增加线条与正文隔离。
“Terminal Display” 格式	“Terminal Display” 格式表示屏幕输出信息。此外，屏幕输出信息中夹杂的用户从终端输入的信息采用加粗字体表示。



## 命令行格式约定

格式	意义
<b>粗体</b>	命令行关键字（命令中保持不变、必须照输的部分）采用加粗字体表示。
<i>斜体</i>	命令行参数（命令中必须由实际值进行替代的部分）采用斜体表示。
[ ]	表示用“[ ]”括起来的部分在命令配置时是可选的。
{ x   y   ... }	表示从两个或多个选项选取一个。
[ x   y   ... ]	表示从两个或多个选项选取一个或者不选。
{ x   y   ... } *	表示从两个或多个选项选取多个，最少选取一个，最多选取所有选项。
[ x   y   ... ] *	表示从两个或多个选项选取多个或者不选。

## 表格内容约定

内容	说明
-	表格中的无内容单元。
*	表格中的内容用户可根据需要进行配置。

## 修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2009-09-30	00B10	优化部分描述。
2009-09-01	00B01	第一次发布。

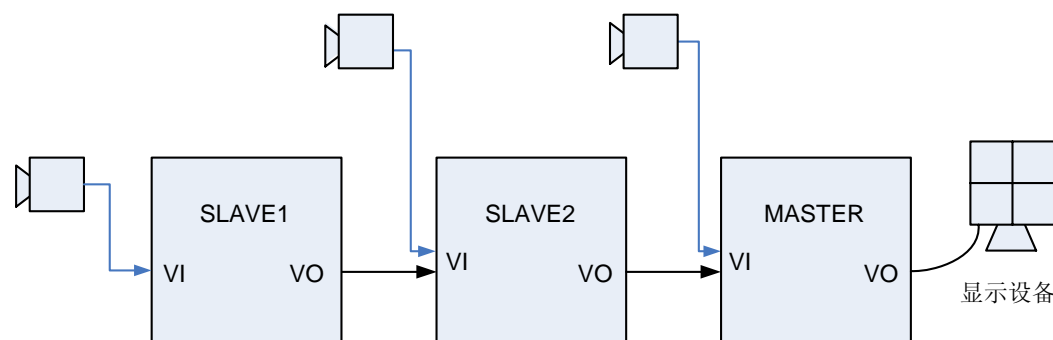


# 1 概述

Hi3520 的视频级联指将多片（若干从片，一个主片）Hi3520 的视频输出图像级联，经过画面合成后在主片的视频输出接口上输出显示。从片通过视频级联，就可将视频图像实时传送到主片，无需借助 PCI 等常规数据传输方式。Hi3520 最多可以容纳 32 路视频画面合成输出，合成画面的最大尺寸 1280%1024。

级联时，从片数可以是 2 片，也可以更多。图 1-1 显示了三片视频级联的连接示意，VO 与 VI 之间通过硬件直连，通过 BT.1120 协议以数字信号进行视频数据的传递。级联时需要占用 VO 的高清设备（HD）及 VI 的 0 和 1 设备。

图1-1 视频级联示意图



级联时，视频数据传递过程如下：

- 步骤 1 从片 1（SLAVE1）将 VI 采集的实时数据或解码数据，直接送给本片 VO 输出。
- 步骤 2 从片 1 的 VO 通过 BT.1120 协议，将数据传递给从片 2 的 VI，VI 将数据采集到片内缓存。
- 步骤 3 从片 2 的 VO 将本片的其他 VI 通道采集的实时数据或解码数据与片内缓存中的数据叠加后，再通过 BT.1120 协议传递给主片 VI。
- 步骤 4 主片 VI 将合成画面采集到内存。此时对主片而言合成画面仅是一路高清视频，与其他 VI 通道采集的数据类似。
- 步骤 5 主片应用程序将本片的其他 VI 通道采集的实时数据或解码数据，与从片 2 传过来的合成画面再次拼接后，送给主片 VO 输出并显示在终端设备上。



----结束





# 2 级联配置

## 2.1 注意事项

Hi3520 的视频级联配置流程中主要有以下四个重要配置项：

- 配置级联主从模式（主片除外）
- 配置画面布局
- 配置级联绑定的通道
- 配置级联使能

在配置过程中，需要注意以下几点：

- Hi3520 的视频级联功能只能在高清通道（HD）上进行。
- 多片视频级联时，主片和从片采用不同的模式：
  - 主片采用非级联模式。
  - 从片采用级联模式，级联模式可分为主模式和从模式。

例如，图 1-1 的三片级联应用中，主片（MASTER）采用非级联模式，从片 1（SLAVE1）和从片 2（SLAVE2）都采用级联模式。从片 1 的 VO 设备是 BT.1120 时序产生的源头，需要配置成级联主模式；从片 2 的 VO 设备继承从片 1 的时序信号，因此只需要配置成级联从模式。



### 说明

只有从片 1 需要配置为级联主模式，其他的从片均需配置为级联从模式。

## 2.2 配置流程

以图 1-1 所示的三片级联为例，详细描述级联配置流程，主要包含以下 3 个方面：

- [从片 1 配置流程](#)
- [从片 2 配置流程](#)
- [主片配置流程](#)

要求级联画面布局为 4 画面平分显示，其中，0、1 画面为从片 1 预览画面；2 画面为从片 2 预览画面；3 画面为主片预览画面。



## 说明

各片的配置流程都是基于 4 应用示例中的 sample 代码进行描述。

## 2.2.1 从片 1 配置流程

从片 1 配置流程如下：

- 步骤 1 配置级联模式为主模式，此时要确保 HD 接口处于禁用状态。
- 步骤 2 配置 HD 输出接口属性，并使能 HD 接口。
- 步骤 3 配置 VI 进行 1D1 的画面采集，该片 VI 的 4 个输入设备都可用。
- 步骤 4 配置 HD 的视频层并使能。
- 步骤 5 配置 HD 的通道属性为四画面平分模式，即 4 个通道，并使能 0、1 通道。
- 步骤 6 绑定 VI 输入通道与 VO 的输出通道 0、1。
- 步骤 7 配置级联画面布局为 4 画面模式，即 pattern 为 4。
- 步骤 8 绑定 HD 的 0 通道与级联位置 0，即 pos 为 0。
- 步骤 9 绑定 HD 的 1 通道与级联位置 1，即 pos 为 1。
- 步骤 10 配置使能级联。

----结束

## 2.2.2 从片 2 配置流程

从片 2 配置流程如下：

- 步骤 1 配置级联模式为从模式，此时要确保 HD 接口处于禁用状态。
- 步骤 2 配置 HD 输出接口属性，并使能 HD 接口。
- 步骤 3 配置 VI 进行 1D1 的画面采集，该片 VI 只有 2、3 设备可用，因为 0、1 设备被高清级联占用。
- 步骤 4 配置 HD 的视频层并使能。
- 步骤 5 配置 HD 的通道属性为四画面平分模式，即 4 个通道，并使能 2 通道。
- 步骤 6 绑定 VI 输入通道与 VO 的输出通道 2。
- 步骤 7 配置级联画面布局为 4 画面模式，即 pattern 为 4。
- 步骤 8 绑定 HD 的 2 通道与级联位置 2，即 pos 为 2。
- 步骤 9 配置使能级联。

----结束

## 2.2.3 主片配置流程

主片配置流程如下：



- 步骤 1 配置 HD 输出接口属性，并使能 HD 接口。
  - 步骤 2 配置 VI 的 0 设备为高清模式，接口时序为 BT.1120 的逐行模式。
  - 步骤 3 配置 VI 的 2 或 3 设备进行 1D1 的画面采集。
  - 步骤 4 使能 VI 的级联功能，主要是用来过滤级联通道的错误图像。
  - 步骤 5 配置 HD 的视频层并使能。
  - 步骤 6 配置 HD 的通道属性为四画面模式，即 4 个通道，并使能 0、3 通道，其中 0 通道的大小是整个显示画面的大小，而其它通道的大小为 4 画面平分模式的大小。
  - 步骤 7 绑定 VI 的 0 设备的 0 通道和 VO 的输出通道 0。
  - 步骤 8 绑定 VI 的 2 或 3 设备的 0 通道和 VO 的输出通道 3。
- 结束





# 3 API 参考

## HI\_MPI\_VO\_EnableCascade

### 【描述】

使能视频级联。

### 【语法】

```
HI_S32 HI_MPI_VO_EnableCascade (HI_VOID);
```

### 【参数】

无。

### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

### 【错误码】

接口返回值	含义
HI_SUCCESS	成功。
HI_ERR_VO_SYS_NOTREADY	系统未准备好。
HI_ERR_VO_DEV_NOT_ENABLE	设备未使能。
HI_ERR_VO_VIDEO_NOT_ENABLE	视频层未使能。

### 【需求】

- 头文件：mpi\_vo.h、hi\_comm\_vo.h



- 库文件：libmpi.a
- 【注意】
- 级联功能只能在高清设备上进行。
  - 级联使能前必须保证高清设备使能和该设备上的视频层使能。
- 【举例】
- 请参见“4 应用示例”。
- 【相关主题】
- [HI\\_MPI\\_VO\\_DisableCascade](#)

HI\_MPI\_VO\_DisableCascade

- 【描述】
- 禁止视频级联。
- 【语法】
- ```
HI_S32 HI_MPI_VO_DisableCascade(HI_VOID);
```
- 【参数】
- 无。
- 【返回值】

| 返回值 | 描述         |
|-----|------------|
| 0   | 成功。        |
| 非 0 | 失败，其值为错误码。 |

【错误码】

| 接口返回值                  | 含义      |
|------------------------|---------|
| HI_SUCCESS             | 成功。     |
| HI_ERR_VO_SYS_NOTREADY | 系统未准备好。 |

- 【需求】
- 头文件：mpi\_vo.h、hi\_comm\_vo.h
  - 库文件：libmpi.a
- 【注意】
- 无。



【举例】

请参见“4 应用示例”。

【相关主题】

[HI\\_MPI\\_VO\\_EnableCascade](#)

## HI\_MPI\_VO\_SetCascadeMode

【描述】

设置视频级联模式。

【语法】

```
HI_S32 HI_MPI_VO_SetCascadeMode(HI_BOOL bSlave);
```

【参数】

| 参数名称   | 描述                                                                                         | 输入/输出 |
|--------|--------------------------------------------------------------------------------------------|-------|
| bSlave | 视频输出级联模式。 <ul style="list-style-type: none"><li>• TRUE：从模式。</li><li>• FALSE：主模式。</li></ul> | 输入    |

【返回值】

| 返回值 | 描述         |
|-----|------------|
| 0   | 成功。        |
| 非 0 | 失败，其值为错误码。 |

【错误码】

| 接口返回值                  | 含义       |
|------------------------|----------|
| HI_SUCCESS             | 成功。      |
| HI_ERR_VO_SYS_NOTREADY | 系统未准备好。  |
| HI_ERR_VO_BUSY         | 视频输出设备忙。 |

【需求】

- 头文件：mpi\_vo.h、hi\_comm\_vo.h
- 库文件：libmpi.a

【注意】



- 设置视频级联模式必须在设备使能之前进行，即必须保证设备已禁止。
- 级联时的第一从片设置为主模式，其它从片设置为从模式。

【举例】

请参见“4 应用示例”。

【相关主题】

[HI\\_MPI\\_VO\\_GetCascadeMode](#)

HI\_MPI\_VO\_GetCascadeMode

【描述】

获取视频级联模式。

【语法】

```
HI_S32 HI_MPI_VO_GetCascadeMode(HI_BOOL *pbSlave);
```

【参数】

| 参数名称    | 描述          | 输入/输出 |
|---------|-------------|-------|
| pbSlave | 视频输出级联模式指针。 | 输出    |

【返回值】

| 返回值 | 描述         |
|-----|------------|
| 0   | 成功。        |
| 非 0 | 失败，其值为错误码。 |

【错误码】

| 接口返回值                  | 含义      |
|------------------------|---------|
| HI_SUCCESS             | 成功。     |
| HI_ERR_VO_SYS_NOTREADY | 系统未准备好。 |
| HI_ERR_VO_NULL_PTR     | 参数空指针。  |

【需求】

- 头文件：mpi\_vo.h、hi\_comm\_vo.h
- 库文件：libmpi.a





【注意】

无。

【举例】

请参见“4 应用示例”。

【相关主题】

[HI\\_MPI\\_VO\\_SetCascadeMode](#)

## HI\_MPI\_VO\_SetCascadePattern

【描述】

设置视频级联画面样式。

【语法】

```
HI_S32 HI_MPI_VO_SetCascadePattern(HI_U32 u32Pattern);
```

【参数】

| 参数名称       | 描述                            | 输入/输出 |
|------------|-------------------------------|-------|
| u32Pattern | 视频级联画面布局样式。<br>取值范围：[0, 127]。 | 输入    |

【返回值】

| 返回值 | 描述         |
|-----|------------|
| 0   | 成功。        |
| 非 0 | 失败，其值为错误码。 |

【错误码】

| 接口返回值                     | 含义      |
|---------------------------|---------|
| HI_SUCCESS                | 成功。     |
| HI_ERR_VO_SYS_NOTREADY    | 系统未准备好。 |
| HI_ERR_VO_INVALID_PATTERN | 无效的样式。  |

【需求】

- 头文件：mpi\_vo.h、hi\_comm\_vo.h
- 库文件：libmpi.a



【注意】

画面布局样式是一个小于 128 的正整数，它的主要作用是检测各片之间传输图像的一致性。即如果要保证图像传输的正确性，就必须保证各片的 pattern 一致。

- 用 pattern 来标识当前画面布局样式，如 4 画面、9 画面、16 画面的 pattern 对应设置为 4、9、16。设置的 patten 值不必和画面布局样式中的画面个数一致，只要保证各片设置的 pattern 值一致即可，如也可以用 1 来表示每个片子上都是 16 分屏的画面布局。
- 使用 pattern 的另一个作用是确保多画面切换时的同步性。当级联的芯片正在进行 4 画面显示时，其中一片先切换到 9 画面，而其它片子陆续切换为 9 画面，如果 pattern 始终不变，画面就会在切换过程中出现错误的几帧。但是如果在 4 画面时 pattern 设置为 4，在 9 画面时 pattern 设置为 9，那么切换过程中出现的错误图像就会被丢弃，保证画面切换的正确性。

【举例】

请参见“4 应用示例”。

【相关主题】

[HI\\_MPI\\_VO\\_GetCascadePattern](#)

HI\_MPI\_VO\_GetCascadePattern

【描述】

获取视频级联画面样式。

【语法】

```
HI_S32 HI_MPI_VO_GetCascadePattern(HI_U32 *pu32Pattern);
```

【参数】

| 参数名称        | 描述            | 输入/输出 |
|-------------|---------------|-------|
| pu32Pattern | 视频级联画面布局样式指针。 | 输出    |

【返回值】

| 返回值 | 描述         |
|-----|------------|
| 0   | 成功。        |
| 非 0 | 失败，其值为错误码。 |

【错误码】



| 接口返回值                  | 含义      |
|------------------------|---------|
| HI_SUCCESS             | 成功。     |
| HI_ERR_VO_SYS_NOTREADY | 系统未准备好。 |

【需求】

- 头文件：mpi\_vo.h、hi\_comm\_vo.h
- 库文件：libmpi.a

【注意】

无。

【举例】

请参见“4 应用示例”。

【相关主题】

[HI\\_MPI\\_VO\\_SetCascadePattern](#)

## HI\_MPI\_VO\_CascadePosBindChn

【描述】

绑定级联区域与视频输出通道。

【语法】

```
HI_S32 HI_MPI_VO_CascadePosBindChn (HI_U32 u32Pos, VO_CHN VoChn);
```

【参数】

| 参数名称   | 描述                         | 输入/输出 |
|--------|----------------------------|-------|
| u32Pos | 视频级联位置编号。<br>取值范围：[0, 31]。 | 输入    |
| VoChn  | 视频层通道号。<br>取值范围：[0, 31]。   | 输入    |

【返回值】

| 返回值 | 描述         |
|-----|------------|
| 0   | 成功。        |
| 非 0 | 失败，其值为错误码。 |



【错误码】

| 接口返回值                      | 含义         |
|----------------------------|------------|
| HI_SUCCESS                 | 成功。        |
| HI_ERR_VO_SYS_NOTREADY     | 系统未准备好。    |
| HI_ERR_VO_DEV_NOT_ENABLE   | 视频输出设备为使能。 |
| HI_ERR_VO_INVALID_POSITION | 无效的级联位置编号。 |
| HI_ERR_VO_INVALID_CHNID    | 通道号无效。     |
| HI_ERR_VO_CHN_NOT_ALLOC    | 通道资源未分配。   |
| HI_ERR_VO_CHN_NOT_CONFIG   | 通道未配置。     |

【需求】

- 头文件：mpi\_vo.h、hi\_comm\_vo.h
- 库文件：libmpi.a

【注意】

- 视频级联位置编号必须在所有的级联芯片上是唯一标识的，也就是级联最多可以输出 32 个通道。
- Pos 的值不能有重复的，如果存在两个或者两个以上通道的 Pos 值相同，那么传输图像就会错误。Pos 相当于给 Chn 取了个别名，所有需要级联的通道都必须有一个名字区别于其它通道，这样才能保证传输图像的正确性。

【举例】

请参见“4 应用示例”。

【相关主题】

[HI\\_MPI\\_VO\\_CascadePosUnBindChn](#)

HI\_MPI\_VO\_CascadePosUnBindChn

【描述】

解绑定级联区域与视频输出通道。

【语法】

```
HI_S32 HI_MPI_VO_CascadePosUnBindChn(HI_U32 u32Pos, VO_CHN VoChn);
```

【参数】



| 参数名称   | 描述                         | 输入/输出 |
|--------|----------------------------|-------|
| u32Pos | 视频级联位置编号。<br>取值范围：[0, 31]。 | 输入    |
| VoChn  | 视频层通道号。<br>取值范围：[0, 31]。   | 输入    |

#### 【返回值】

| 返回值 | 描述         |
|-----|------------|
| 0   | 成功。        |
| 非 0 | 失败，其值为错误码。 |

#### 【错误码】

| 接口返回值                      | 含义         |
|----------------------------|------------|
| HI_SUCCESS                 | 成功。        |
| HI_ERR_VO_SYS_NOTREADY     | 系统未准备好。    |
| HI_ERR_VO_DEV_NOT_ENABLE   | 视频输出设备未使能。 |
| HI_ERR_VO_INVALID_POSITION | 无效的级联位置编号。 |
| HI_ERR_VO_INVALID_CHNID    | 通道号无效。     |
| HI_ERR_VO_CHN_NOT_ALLOC    | 通道资源未分配。   |
| HI_ERR_VO_CHN_NOT_CONFIG   | 通道未配置。     |

#### 【需求】

- 头文件：mpi\_vo.h、hi\_comm\_vo.h
- 库文件：libmpi.a

#### 【注意】

视频级联位置编号必须在所有的级联芯片上是唯一标识的，也就是级联最多可以输出 32 个通道。

#### 【举例】

请参见“[4 应用示例](#)”。

#### 【相关主题】



## HI\_MPI\_VO\_CascadePosBindChn



# 4 应用示例

Hi320 视频级联的功能实现需要在各个片上进行不同的配置，其中三片级联最具有代表性和可扩展性。主片处于非级联模式，从片第一片采用级联主模式，其它从片都采用级联从模式，若进行三片以上的级联，只需复制三片级联用例中的第二片配置即可。下面提供三片级联时各片的代码片段供参考：

- [从片 1 代码示例](#)
- [从片 2 代码示例](#)
- [主片代码示例](#)

## 4.1 从片 1 代码示例

```
/******  
*****  
* Function      :   Sample_Cascade_Slave1  
* Description   :   级联演示用例 从片1  
* Title        :   级联四画面预览从片1 VGA 1280x1024  
* Criticality   :   H  
* Pre-Condition :   三片级联，从1->从2->主  
* Input        :   2D1 预览图像  
* Expected     :   从片1图像显示在四画面的第1和第2通道  
* Remark       :  
* Date        :  
*****  
*****/  
HI_S32 Sample_Cascade_Slave1(HI_VOID)  
{  
    HI_U32 i;  
    HI_S32 s32Ret;  
    HI_U32 u32Pos = 0;  
    HI_U32 u32Pattern = 4;
```



```
HI_U32 u32Width = 1024, u32Height = 768;
HI_U32 u32HalfWidth = u32Width/2, u32HalfHeight = u32Height/2;
VO_INTF_SYNC_E mode = VO_OUTPUT_1024x768_60;

VO_CHN_ATTR_S stChnAttr[4] = {{1,
{0,0,u32HalfWidth,u32HalfHeight},HI_TRUE, HI_FALSE},
{1, {u32HalfWidth,0,u32HalfWidth,u32HalfHeight},HI_TRUE,
HI_FALSE},
{1, {0,u32HalfHeight,u32HalfWidth,u32HalfHeight},HI_TRUE,
HI_FALSE},
{1,
{u32HalfWidth,u32HalfHeight,u32HalfWidth,u32HalfHeight},HI_TRUE,
HI_FALSE}}};

VO_VIDEO_LAYER_ATTR_S stLayerAttr =
{{0,0,u32Width,u32Height},{u32Width,u32Height},
25,PIXEL_FORMAT_YUV_SEMIPLANAR_422,VO_DEFAULT_CHN}};

TITLE("CASCADE SLAVE1: 4xCIF MODE");

/* to ensure HD interface is disabled */
(HI_VOID)HI_MPI_VO_Disable(HD);

/* set cascade mode */
s32Ret = HI_MPI_VO_SetCascadeMode(HI_FALSE);
if (s32Ret != HI_SUCCESS)
{
    printf("set cascade mode faild with errno %#x\n", s32Ret);
    return HI_FAILURE;
}

/* configure and enalbe VO HD interface */
if (HI_SUCCESS != SetVouDev(HD, VO_BKGRD_BLUE, VO_CCD_SLAVE_INTF,
mode))
{
    printf("Enalbe vo hd interface failed!\n");
    return HI_FAILURE;
}

/* configure and enable VI device 0, 1 */
if (HI_SUCCESS != StartViDev12(2, VI_SIZE_D1,
PIXEL_FORMAT_YUV_SEMIPLANAR_422, VI_MODE_BT656,
VI_WORK_MODE_4D1))
```





```
{
    printf("Enable vi device 0, 1 failed!\n");
    return HI_FAILURE;
}

/* configure and enable HD video layer */
if (HI_SUCCESS != StartVideoLayer(HD, &stLayerAttr))
{
    printf("Enalbe HD video layer failed!\n");
    return HI_FAILURE;
}

/* configure and enable vo HD channel 0, 1 */
for (i = 0 ; i < 2 ; i++)
{
    if (HI_SUCCESS != HI_MPI_VO_SetChnAttr(HD, i, &stChnAttr[i]))
    {
        printf("Set channel %d attribute failed!\n", i);
        return HI_FAILURE;
    }

    if (HI_SUCCESS != HI_MPI_VO_EnableChn(HD, i))
    {
        printf("Enable channel %d attribute failed!\n", i);
        return HI_FAILURE;
    }

    if (HI_SUCCESS != HI_MPI_VI_BindOutput(0, i, HD, i))
    {
        printf("vi bind vo channel %d attribute failed!\n", i);
        return HI_FAILURE;
    }
}

/* set cascade pattern */
s32Ret = HI_MPI_VO_SetCascadePattern(u32Pattern);
if (s32Ret != HI_SUCCESS)
{
    printf("set cascade pattern faild with errno %#x\n", s32Ret);
    return HI_FAILURE;
}

/* set cascade postion */
for (i = 0 ; i < 2 ; i++)
```



```

    {
        s32Ret = HI_MPI_VO_CascadePosBindChn(i, i);
        if (s32Ret != HI_SUCCESS)
        {
            printf("set cascade position failed with errno %#x\n", s32Ret);
            return HI_FAILURE;
        }
    }

    /* enable cascade */
    s32Ret = HI_MPI_VO_EnableCascade();
    {
        printf("enable cascade failed with errno %#x\n", s32Ret);
        return HI_FAILURE;
    }

    while(1);

    return HI_SUCCESS;
}

```

## 4.2 从片 2 代码示例

```

/*****
*****
* Function      :   Sample_Cascade_Slave2
* Description   :   级联演示用例 从片2
* Title        :   级联四画面预览从片1 VGA 1280x1024
* Criticality   :   H
* Pre-Condition :   三片级联，从1->从2->主
* Input        :   1D1 预览图像
* Expected     :   从片2图像显示在四画面的第3通道
* Remark       :
* Date        :
*****
*****/
HI_S32 Sample_Cascade_Slave2(HI_VOID)
{
    HI_U32 i;
    HI_S32 s32Ret;
    HI_U32 u32Pos = 2;
    HI_U32 u32Pattern = 4;

```



```
HI_U32 u32Width = 1024, u32Height = 768;
HI_U32 u32HalfWidth = u32Width/2, u32HalfHeight = u32Height/2;
VO_INTF_SYNC_E mode = VO_OUTPUT_1024x768_60;

VO_CHN_ATTR_S stChnAttr[4] = {{1,
{0,0,u32HalfWidth,u32HalfHeight},HI_TRUE, HI_FALSE},
{1, {u32HalfWidth,0,u32HalfWidth,u32HalfHeight},HI_TRUE, HI_FALSE},
{1, {0,u32HalfHeight,u32HalfWidth,u32HalfHeight},HI_TRUE,
HI_FALSE},
{1,
{u32HalfWidth,u32HalfHeight,u32HalfWidth,u32HalfHeight},HI_TRUE,
HI_FALSE}}};

VO_VIDEO_LAYER_ATTR_S stLayerAttr =
{{0,0,u32Width,u32Height},{u32Width,u32Height},

25,PIXEL_FORMAT_YUV_SEMIPLANAR_422,VO_DEFAULT_CHN}};

TITLE("CASCADE SLAVE2: 4xCIF MODE");

/* to ensure HD interface is disabled */
(HI_VOID)HI_MPI_VO_Disable(HD);

/* set cascade mode */
s32Ret = HI_MPI_VO_SetCascadeMode(HI_TRUE);
if (s32Ret != HI_SUCCESS)
{
    printf("set cascade mode faild with errno %#x\n", s32Ret);
    return HI_FAILURE;
}

/* configure and enalbe VO HD interface */
if (HI_SUCCESS != SetVouDev(HD, VO_BKGRD_BLUE, VO_CCD_SLAVE_INTF,
mode))
{
    printf("Enalbe vo hd interface failed!\n");
    return HI_FAILURE;
}

/* configure and enable VI device 2, 3 */
if (HI_SUCCESS != StartViDev34(1, VI_SIZE_D1,
PIXEL_FORMAT_YUV_SEMIPLANAR_422, VI_MODE_BT656,
VI_WORK_MODE_4D1))
```



```
{
    printf("Enable vi device 0, 1 failed!\n");
    return HI_FAILURE;
}

/* configure and enable HD video layer */
if (HI_SUCCESS != StartVideoLayer(HD, &stLayerAttr))
{
    printf("Enalbe HD video layer failed!\n");
    return HI_FAILURE;
}

/* configure and enable vo HD channel 2 */
if (HI_SUCCESS != HI_MPI_VO_SetChnAttr(HD, 2, &stChnAttr[2]))
{
    printf("Set channel 2 attribute failed!\n");
    return HI_FAILURE;
}

if (HI_SUCCESS != HI_MPI_VO_EnableChn(HD, 2))
{
    printf("Enable channel 2 attribute failed!\n");
    return HI_FAILURE;
}

if (HI_SUCCESS != HI_MPI_VI_BindOutput(2, 0, HD, 2))
{
    printf("vi bind vo channel 2 attribute failed!\n");
    return HI_FAILURE;
}

/* set cascade pattern */
s32Ret = HI_MPI_VO_SetCascadePattern(u32Pattern);
if (s32Ret != HI_SUCCESS)
{
    printf("set cascade pattern faild with errno %#x\n", s32Ret);
    return HI_FAILURE;
}

/* set cascade postion */
s32Ret = HI_MPI_VO_CascadePosBindChn(2, 2);
if (s32Ret != HI_SUCCESS)
{
    printf("set cascade position faild with errno %#x\n", s32Ret);
}
```



```
        return HI_FAILURE;
    }

    /* enable cascade */
    s32Ret = HI_MPI_VO_EnableCascade();
    {
        printf("enable cascade faild with errno %x\n", s32Ret);
        return HI_FAILURE;
    }

    while(1);

    return HI_SUCCESS;
}
```

### 4.3 主片代码示例

```
/******
 * Function      : Sample_Cascade_Master
 * Description    : 级联演示用例 主片
 * Title         : 级联四画面预览主片 VGA 1280x1024
 * Criticality    : H
 * Pre-Condition : 三片级联, 从1->从2->主
 * Input         : 1D1 预览图像
 * Expected       : 主片图像显示在四画面的第4通道
 * Remark        :
 * Date          :
 *****/
HI_S32 Sample_Cascade_Master(HI_VOID)
{
    HI_U32 i;
    HI_S32 s32Ret;

    HI_U32 u32Width = 1024, u32Height = 768;
    HI_U32 u32HalfWidth = u32Width/2, u32HalfHeight = u32Height/2;
    VO_INTF_SYNC_E mode = VO_OUTPUT_1024x768_60;

    /* note: channel 0 is different with others */
    VO_CHN_ATTR_S stChnAttr[4] = {{1, {0,0,u32Width,u32Height},HI_TRUE,
    HI_FALSE},
        {1, {u32HalfWidth,0,u32HalfWidth,u32HalfHeight},HI_TRUE, HI_FALSE},
        {1, {0,u32HalfHeight,u32HalfWidth,u32HalfHeight},HI_TRUE,
```



```
HI_FALSE},
    {1,
    {u32HalfWidth,u32HalfHeight,u32HalfWidth,u32HalfHeight},HI_TRUE,
    HI_FALSE}};

    VO_VIDEO_LAYER_ATTR_S stLayerAttr =
    {{0,0,u32Width,u32Height},{u32Width,u32Height},

    25,PIXEL_FORMAT_YUV_SEMIPLANAR_422,VO_DEFAULT_CHN}};

    TITLE("CASCADE MASTER: 4xCIF MODE");

    /* start vi HD channel as cascade receiver */
    s_enViInputMode = VI_MODE_BT1120_PROGRESSIVE;
    if (HI_SUCCESS != StartViHD(gs_u32CscdChnCnt, u32Width, u32Height,
                                60, gs_u32CscdTarFrmRate))
    {
        printf("start vi hd mode failed!\n");
        return HI_FAILURE;
    }

    /* configure and enable VI device 2, 3 for local video input */
    if (HI_SUCCESS != StartViDev34(1, VI_SIZE_D1,
                                    PIXEL_FORMAT_YUV_SEMIPLANAR_422, VI_MODE_BT656,
    VI_WORK_MODE_4D1))
    {
        printf("Enable vi device 0, 1 failed!\n");
        return HI_FAILURE;
    }

    /* configure and enable VO HD interface */
    if (HI_SUCCESS != SetVouDev(HD, VO_BKGRD_BLUE, VO_CCD_SLAVE_INTF,
    mode))
    {
        printf("Enalbe vo hd interface failed!\n");
        return HI_FAILURE;
    }

    /* configure and enable HD video layer */
    if (HI_SUCCESS != StartVideoLayer(HD, &stLayerAttr))
    {
        printf("Enalbe HD video layer failed!\n");
        return HI_FAILURE;
    }
}
```



```
/* configure and enable vo HD channel 0 to receive cascade picture */
if (HI_SUCCESS != HI_MPI_VO_SetChnAttr(HD, 0, &stChnAttr[0]))
{
    printf("Set channel 2 attribute failed!\n");
    return HI_FAILURE;
}
if (HI_SUCCESS != HI_MPI_VO_EnableChn(HD, 0))
{
    printf("Enable channel 2 attribute failed!\n");
    return HI_FAILURE;
}

/* configure and enable vo HD channel 3 to overlap local picture */
if (HI_SUCCESS != HI_MPI_VO_SetChnAttr(HD, 3, &stChnAttr[3]))
{
    printf("Set channel 2 attribute failed!\n");
    return HI_FAILURE;
}
if (HI_SUCCESS != HI_MPI_VO_EnableChn(HD, 3))
{
    printf("Enable channel 2 attribute failed!\n");
    return HI_FAILURE;
}

/* bind hd channel 0 with vi channel 0 to receive cascade picture */
if (HI_SUCCESS != HI_MPI_VI_BindOutput(0, 0, HD, 0))
{
    printf("vi bind vo channel 2 attribute failed!\n");
    return HI_FAILURE;
}

/* bind hd channel 3 to overlap local picture */
if (HI_SUCCESS != HI_MPI_VI_BindOutput(2, 0, HD, 3))
{
    printf("vi bind vo channel 2 attribute failed!\n");
    return HI_FAILURE;
}

while(1);

return HI_SUCCESS;
}
```