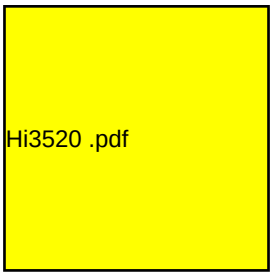


Hi3520 图形开发

用户指南

文档版本 00B20

发布日期 2009-10-30

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

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

插图目录

表格目录

前 言

概述

本文为图形开发推荐了 2 个方案，分别从方案介绍、衍生方案、开发流程、应用场景及优点和限制介绍，为用户在进行图形开发时提供参考。

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
粗体	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。
斜体	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中选取一个。
[x | y | ...]	表示从两个或多个选项中选取一个或者不选。
{ x | y | ... } *	表示从两个或多个选项中选取多个，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中选取多个或者不选。

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
2009-10-30	00B20	修改了各个图形的为 16 通道分割线,，并修改了部分文字描述。
2009-09-30	00B10	基线版本。
2009-09-02	00B01	第一次发布。

1 图形层介绍

1.1 概述

海思数字媒体处理平台提供一整套机制支持图形界面的开发，主要包括：

- 图形二维加速引擎（Two Dimensional Engine，简称 TDE），它利用硬件加速对图形图像进行处理。
- Hisilicon Framebuffer（以下简称 HiFB）用于管理叠加图形层，它不仅提供 Linux Framebuffer 的基本功能，还在 Linux Framebuffer 的基础上增加层间 colorkey、层间 Alpha 等扩展功能。

Hi3520 芯片支持 5 个图形层，可以分别在 3 个 VO 输出设备上实现后端 OSD、GUI 界面、鼠标功能。

说明
- TDE 相关使用方法请参见《Hi3520 TDE API 参考》
- HiFB 相关使用方法请参见《HiFB 开发指南》和《HiFB API 参考》

1.2 图形层体系结构

Hi3520 芯片支持 3 个 VO 输出设备：高清输出设备（简称 HD）、辅助显示设备（简称 AD）、标清输出设备（简称 SD）。

说明
每个输出设备支持的接口类型和时序请参见《Hi3520 H.264 编解码处理器用户指南》的“6.2 VOU 章节”。

5 个图形层（简称 G0～G4）与各设备有一定的约束关系，如表 1-1 所示。

表1-1 图形层与输出设备的对应关系表

图形层	对应显示设备	说明
叠加图行层 0（G0）	HD	G0 只能在 HD 设备上显示。

图形层	对应显示设备	说明
叠加图行层 1（G1）	HD 或 AD	G1 可在 HD 或 AD 设备上显示，调用者可通过绑定接口指定显示设备。 G1 切换为静态切换，即 HD 和 AD 设备会自动关闭再开启。G1 切换的用户操作如下：先关闭 G1→设置新的绑定关系→配置并开启 G1。
叠加图行层 2（G2）	AD	G2 只能在 AD 设备上显示。
叠加图行层 3（G3）	SD	G3 只能在 SD 设备上显示。
叠加图行层 4（G4）	HD 或 AD	G4 常用作鼠标显示。 G4 可在 HD 或 AD 设备上显示，调用者可通过绑定接口指定显示设备。 G4 切换的用户操作和 G1 切换时一致，即先关闭 G4→设置新的绑定关系→配置并开启 G4。 G4 切换与 G1 切换的不同点是：G4 切换时 HD 和 AD 设备状态保持不变，不会关闭再开启，故不会出现短暂黑屏现象；而 G1 在用户设置绑定关系时，会自动关闭再开启 HD 和 AD 设备，故会出现短暂黑屏。 G4 总是处在显示设备叠加层的最高层。如 HD 上有视频层、G0 和 G4，则叠加顺序为 G0 在视频层之上、G4 在 G0 之上。

说明

为了显示图形层，用户必须先设定 G1 和 G4 的绑定关系，然后配置并启动输出设备（通过 VOU 模块的接口），最后通过 HiFB 模块接口操作图像层使之显示。通过指定图形层到设备的绑定关系，G1 和 G4 可分别或同时绑定到 HD 或 AD 设备上。

2 图形开发推荐方案

2.1 概述

在监控领域中，一般输出设备的图形用户界面内容包括：

- 后端 OSD：显示画面分割线、通道号、时间等信息，用以界定多画面显示布局。
- GUI 界面：包括各种菜单、进度条等元素，用户通过操作 GUI 界面进行设备配置。
- 鼠标：提供更方便易用的界面菜单操作方式。

以上 3 类图形内容可以通过 1 个图形层实现，也可以通过多个图形层实现。Hi3520 芯片提供 5 个图形层，为了指导用户正确、合理、有效的利用这些图层，以满足不同的输出界面应用场景，在此推荐几种方案以供参考。

2.2 单图层实现用户界面方案

2.2.1 方案介绍

该方案总体思路是：每个设备都只使用 1 层图形层来完成本设备的后端 OSD、GUI 和鼠标的显示，鼠标也可单独使用 G4 图像层来实现。

可具体描述为：每个输出设备使用一个图形层来完成本设备的后端 OSD、GUI；GUI 画在独立的缓存上，后端 OSD 直接画在 FB 显存中，再通过 TDE 进行 alpha 混合；鼠标可以使用单独的 G4 图形层，也可以跟 OSD、GUI 共用一个图层，共用图层的时候，可以画在 GUI 缓存上。

该方案使用了以下机制：

- 每个设备的后端 OSD 直接绘制在各自的 FB 显存中。

  如在 HD 的图形层 G0 对应的 FB 显存中绘制 16 画面分割布局，AD 的 G2 图层对应的 FB 中绘制 4 画面分割布局，SD 的 G3 图层对应的 FB 中绘制通道号信息。
- 每个设备一块 GUI 画布，GUI 变更时局部刷新。

  每个设备使用一块独立的缓存绘制 GUI（称该块缓存为 GUI 画布），当 GUI 变更时仅需要进行局部刷新。

- GUI 画布整体搬移至相应图层的 FB 显存中

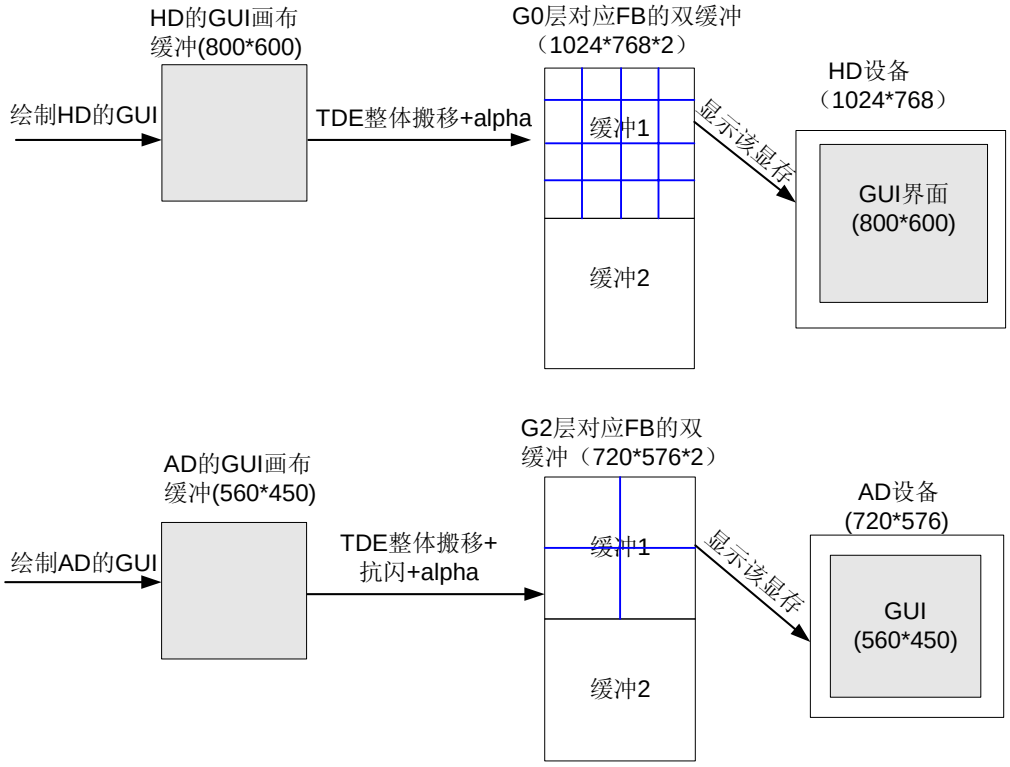
  将绘制好的画布整体搬移到相应的 FB 缓冲中，在此过程中可利用 TDE 实现 GUI 和 OSD 的叠加透明效果。每次 GUI 或 OSD 有变动时，由于是对画布和 OSD 整体做叠加，故不需要针对局部信息计算 GUI 和 OSD 的叠加区域。

- FB 双缓冲

  为防止一块 FB 缓冲被边绘制边显示而导致绘制过程可见，推荐使用 FB 双缓冲机制。即 FB 分配 2 块大小相同的缓冲作为显存交替绘制和显示。如 VO 正在显示缓冲 2，则本次绘制的对象为缓冲 1，然后通过 FB 的 PAN_DISPLAY 系统调用通知 VO 显示缓冲 1。

方案的结构如图 2-1 所示。

图2-1 单图层方案的结构示意图



该方案在后端 OSD 或者 GUI 界面变动时，都需要重新绘制 FB 缓存：

- 本设备的后端 OSD 改变时，如 16 通道分割线切换到 9 通道分割线：先清空 FB 缓存，再绘制新的 OSD，再将 GUI 界面整体搬移到 FB 缓存中。
- GUI 界面每次变动时，都需要先清空 FB 缓存，再绘制 OSD，然后将新的 GUI 界面整体搬移到 FB 缓存中。

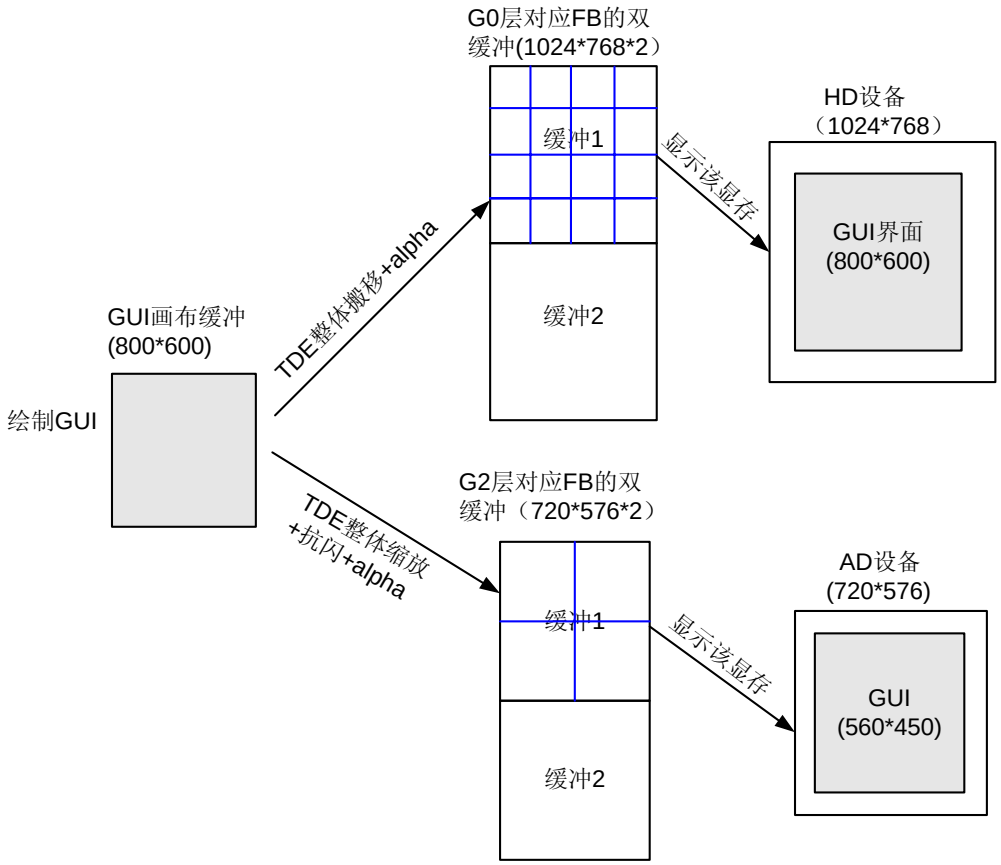
2.2.2 衍生方案

当 AD 和 HD 设备上同时显示同样的 GUI 界面时，该方案可简化仅有一块 GUI 画布缓存：

- 画布大小与 HD 的 GUI 层大小相同（800*600），用户可按照 HD 的 GUI 规格（如 800*600）准备一套图片，每次 GUI 变更时仅局部绘制画布，而 AD 的 GUI 则是将画布整体经过缩放、抗闪得到，其效果略差于 HD 上的 GUI。
- 每次更新画布后，对于 HD 设备，由于画布大小与 GUI 界面大小相同，故利用 TDE 做整体搬移操作即可；对于 AD 设备，需要利用 TDE 对画布整体进行缩放至 G2 的 FB 显存中，同时进行抗闪烁处理（因 AD 是隔行设备）。

该衍生方案的结构如图 2-2 所示。

图2-2 衍生方案的结构图



2.2.3 开发流程

方案 1 的开发流程

以 HD 和 AD 设备上的 GUI 和 OSD 为例：HD 设备上 16 画面等分分割线，AD 设备上 4 画面等分分割线，且 HD 和 AD 同时显示同样的 GUI。

若此时 GUI 界面有变化，则该方案的实现过程为：

步骤 1 清空 G0 和 G2 对应 FB 的空闲缓冲（假设为缓冲 1，缓冲 2 正在被 VO 显示）。

步骤 2 在 G0 对应的 FB 缓冲 1 中绘制 16 通道分割线。

步骤 3 在 G2 对应的 FB 缓冲 1 中绘制 4 通道分割线。

步骤 4 局部更新画布。

步骤 5 用 TDE 将画布整体搬移到 G0 缓冲 1 的合适位置，此过程可以做 alpha 透明度叠加以实现 GUI 半透明效果。

步骤 6 用 TDE 将画布整体缩放到 G2 缓冲 1 的合适位置，此过程可以做抗闪、alpha 透明度叠加（以实现 GUI 半透明效果）。

步骤 7 通过 FB 接口调用 PAN_DISPLAY 通知 HD 显示已准备好的 G0 缓冲 1。

步骤 8 通过 FB 接口调用 PAN_DISPLAY 通知 AD 显示已准备好的 G2 缓冲 1。

----结束

2.2.4 应用场景

应用场景如下：

- 每个设备上有各自的后端 OSD（如 HD 为 16 画面分割布局，AD 为 4 画面分割布局，SD 为单画面但有通道号信息）。
- 2 或多个输出设备上同时有 GUI 界面（相同或者不同）。

2.2.5 优点和限制

该方案具有以下优点：

- 可同时在多个设备上显示 GUI 界面。
- GUI 画布可局部刷新，节省总线带宽和 TDE 性能。
- 可实现 GUI 和 OSD 的叠加透明效果，且用户控制流程简单。每次 GUI 或 OSD 有变动时，由于是对画布和 OSD 整体做叠加，故不需要针对局部信息计算 GUI 和 OSD 的叠加区域。
- 对于衍生方案，用户仅需要一套 GUI 界面的图片，就可适应不同分辨率设备的 GUI 需求，节省 Flash 空间。

该方案具有以下约束：

- 对于衍生方案：AD/SD 上的 GUI 是画布缩放得到的，故效果略差于 HD 的 GUI。

2.3 多个图层实现用户界面方案

2.3.1 方案介绍

本方案利用 3 个图形层分别实现 OSD、GUI 界面、鼠标。

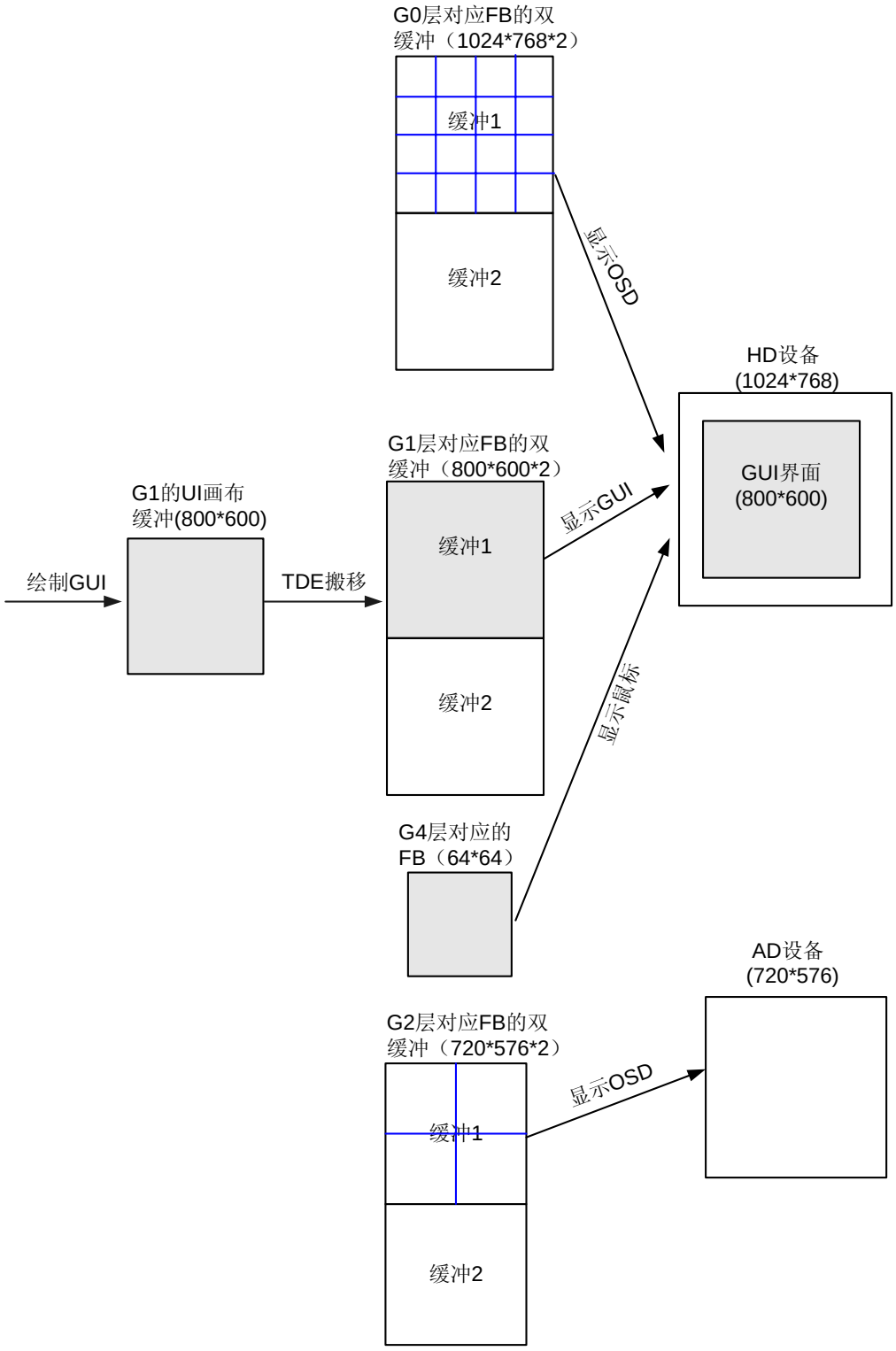
每个设备都只使用 1 层图形层来实现本设备的 OSD，即 HD 使用 G0 层，AD 使用 G2 层，SD 使用 G3 层实现本设备的后端 OSD。

GUI 使用图形层 G1 实现。当 GUI 界面移到某个设备上时，如移到 HD 设备上，HD 同时绑定两个图像层 G0 和 G1，且利用图形层间 alpha 叠加可实现 GUI 半透明。

当 GUI 界面从 HD 移到 AD 上时，先关闭 G1，再通过设置绑定关系将 G1 绑定到新的设备 AD 上，最后通过 HiFB 接口并开启 G1 层即可。

鼠标利用 G4 层实现，G4 层大小为鼠标图片大小。鼠标移动时，通过设置 G4 对应的 FB 的显示位置实现。鼠标可以动态的在 HD 和 AD 间移动，也需要用户先关闭 G4 层，再通过设置绑定关系将 G4 绑定到新的设备上，最后并开启 G4 层。

图2-3 HD 上显示 OSD、GUI 和鼠标的结构示意图



2.3.2 衍生方案

由于切换 GUI 图层 G1 时需要关闭 HD 和 AD 设备，导致短暂黑屏现象。在此推荐衍生方案以实现 GUI 的动态切换。

该衍生方案利用两个图形层分别实现 OSD 和 GUI+鼠标界面：

每个设备都使用 1 层图形层来完成本设备的 OSD，即 HD 使用 G0 层，AD 使用 G2 层，SD 使用 G3 层实现本设备的后端 OSD。

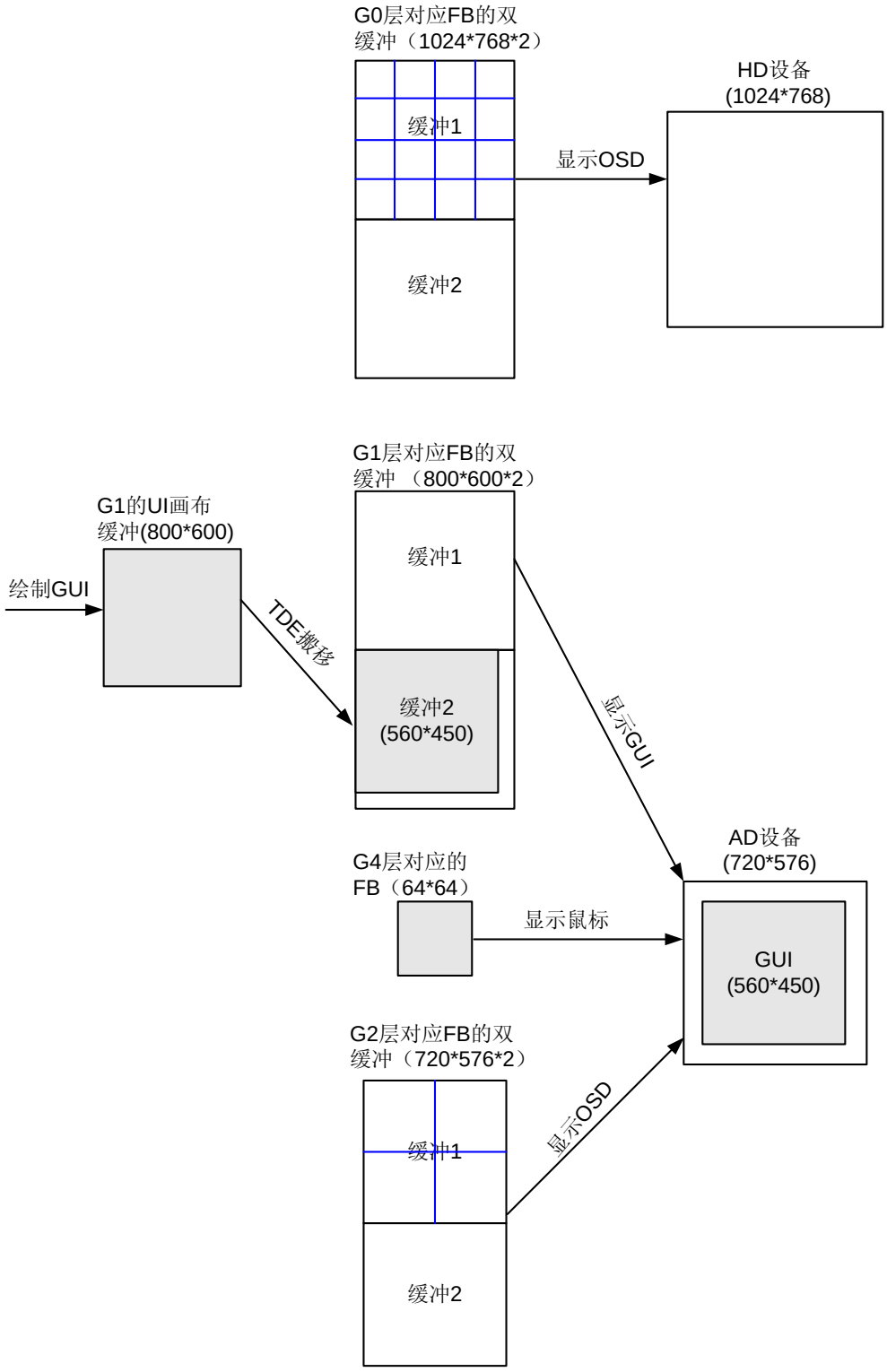
GUI 和鼠标都使用图形层 G4 实现。由于 Hi3520 芯片支持 G4 动态的在 HD 和 AD 间切换，即切换 G4 时 HD 和 AD 不需要关闭再开启，故没有短暂黑屏现象。

说明

- 当 GUI 界面移到某个设备上时，如移到 HD 设备上，HD 同时绑定两个图像层 G0 和 G4，且利用图形层间 alpha 叠加可实现 GUI 半透明。
- 当 GUI 界面从 HD 移到 AD 上时，将 G4 关闭，然后绑定到 AD 上，再开启 G4 层。
- 鼠标移动时，仅局部刷新 G4 的画布即可。

GUI 和鼠标切换至 AD 上的结构如图 2-4 所示。

图2-4 GUI 和鼠标切换至 AD 上的结构示意图



2.3.3 开发流程

以 HD 和 AD 设备上的 GUI 和 OSD 为例：HD 设备上 16 画面等分分割线，AD 设备上 4 画面等分分割线，GUI 在 HD 和 AD 间切换。

方案 2 的开发流程

以 GUI 界面从 HD 切换到 AD 上为例，配置步骤如下：

步骤 1 局部更新 G1 对应的画布。

步骤 2 关闭 G1 层。

步骤 3 设置 G1 绑定到 AD 设备。

步骤 4 通过 HiFB 接口设置 G1 属性并开启 G1 层：用 TDE 将画布整体缩放到 G1 缓冲 2 的合适位置。注意：由于 AD 上的 GUI 界面大小小于画布大小，故需要将画布缩放到和 AD 的 GUI 大小一样，即改变 G1 对应的 FB 的显示分辨率 560*450，以适应 AD 上的 GUI 界面大小）。

----结束

衍生方案的开发流程

以 GUI 界面从 HD 切换到 AD 上为例，配置步骤如下：

步骤 1 局部更新画布。

步骤 2 关闭 G4 层。

步骤 3 设置 G4 绑定到 AD 设备。

步骤 4 通过 HiFB 接口设置 G4 属性并开启 G4 层：用 TDE 将画布整体缩放到 G4 缓冲 2 的合适位置。**注意**：由于 AD 上的 GUI 界面大小小于画布大小，故需要将画布缩放到和 AD 的 GUI 大小一样。

----结束

2.3.4 应用场景

应用场景如下：

- 3 个输出设备有各自的后端 OSD（如 HD 为 16 画面分割布局，AD 为 4 画面分割布局，SD 为单画面但有通道号信息）
- 某一时刻仅有一个设备显示 GUI，且要求 GUI 能在 AD 和 HD 上切换（动态或静态地切换）。
- 某一时刻仅有一个设备上显示鼠标，且要求鼠标能在 AD 和 HD 上动态切换。

2.3.5 优点和限制

该方案具有以下优点：

- GUI 界面和后端 OSD 使用不同的图层，其中一个变化时不需要更新另一个图层，用户控制简单。
- 鼠标可以在 HD 和 AD 设备间动态切换，切换时不需要关闭输出设备。
- 衍生方案支持 GUI 在 HD 和 AD 间动态切换，用户感受增强。
- 用户仅需要一套 GUI 界面的图片，可适应不同分辨率设备的 GUI 需求。
- 一块 GUI 画布，可局部刷新。

该方案具有以下限制：

- 由于 GUI 和 OSD 使用了 2 个图层，相对于方案 1 占用了更多的系统带宽。
- 若使用 G1 做 GUI 界面，GUI 不支持在 HD 和 AD 间动态切换。
- G4 和 G1 不能绑定到 SD 上，故此方案 SD 上不能实现 GUI 界面和鼠标。