



CV186AH TDE API 使用手册

Version: 1.0.1

Release date: 2023/12

©2022 北京晶视智能科技有限公司
本文件所含信息归北京晶视智能科技有限公司所有。
未经授权，严禁全部或部分复制或披露该等信息。

目录

1	声明	2
2	功能概述	3
2.1	目的	3
2.2	适用范围	3
2.3	术语和定义	4
3	API 参考	5
3.1	API 概述	5
3.2	功能函数参考	6
4	数据类型	40
4.1	映射表	40
4.2	详细描述	41
5	错误码	70
5.1	测试	70

修订记录

Revision	Date	Description
1.0.0	2023/08/15	初稿
1.0.1	2023/12/01	更新最新 API 列表

1 声明



法律声明

本数据手册包含北京晶视智能科技有限公司（下称“晶视智能”）的保密信息。未经授权，禁止使用或披露本数据手册中包含的信息。如您未经授权披露全部或部分保密信息，导致晶视智能遭受任何损失或损害，您应对因之产生的损失/损害承担责任。

本文件内信息如有更改，恕不另行通知。晶视智能不对使用或依赖本文件所含信息承担任何责任。本数据手册和本文件所含的所有信息均按“原样”提供，无任何明示、暗示、法定或其他形式的保证。晶视智能特别声明未做任何适销性、非侵权性和特定用途适用性的默示保证，亦对本数据手册所使用、包含或提供的任何第三方的软件不提供任何保证；用户同意仅向该第三方寻求与此相关的任何保证索赔。此外，晶视智能亦不对任何其根据用户规格或符合特定标准或公开讨论而制作的可交付成果承担责任。

联系我们

地址 北京市海淀区丰豪东路 9 号院中关村集成电路设计园（ICPARK）1 号楼

深圳市宝安区福海街道展城社区会展湾云岸广场 T10 栋

电话 +86-10-57590723 +86-10-57590724

邮编 100094（北京）518100（深圳）

官方网站 <https://www.sophgo.com/>

技术论坛 <https://developer.sophgo.com/forum/index.html>

2 功能概述

本文档主要用于指导用户 CVI_TDE 模块开发以便提高开发效率；我们不保证此文档更改后能及时告知用户，因此请使用最新版 released SDK 里面的文档

2.1 目的

本文档主要用于指导用户 CVI_TDE 模块开发以便提高开发效率

2.2 适用范围

此文档描述了 CVI_TDE 模块的软件设计，适用于技术支持工程师、软件开发工程师等

2.3 术语和定义

表 2.1: 术语和定义列表

序号	术语	定义说明
1.	TDE	Two Dimension Engine 二维引擎；本文档中表示硬件上的二维图像渲染引擎。
2.	Blt	Bit Block Transfer 位块传输；其是一种图像处理操作，通常用于在计算机图形中进行位图之间的块传输。这种操作涉及从一个图像区域复制一块位图数据到另一个图像区域，以实现图像合成、转换、缩放、平铺等效果。
3.	Path	路径；其用于描述图形的轮廓和形状。
4.	Stroke	描边；其是将路径的边缘轮廓绘制成线条的操作。通过指定描边属性，您可以控制描边的样式、宽度、颜色等。
5.	Blend	混合；其是将两个图像或颜色按照一定的规则进行合成，以产生最终的效果。
6.	Filter	滤镜；一种通过在图像上应用不同的算法或操作，来改变图像的外观或质量的过程。

3 API 参考

3.1 API 概述

CVI_TDE 是一个轻量级的 2D 图形 API，用于控制 graphic 实现加速矢量和光栅操作。

这里提供以下 API:

TDE 控制:

- `cvi_tde_open`: 打开 TDE 设备。
- `cvi_tde_close`: 关闭 TDE 设备。
- `cvi_tde_begin_job`: 创建 1 个 TDE 任务。
- `cvi_tde_end_job`: 提交添加操作完成的 TDE 任务。
- `cvi_tde_quick_fill`: 向任务中添加快速填充操作。
- `cvi_tde_draw_corner_box`: 向任务中添加绘制复数角框操作。
- `cvi_tde_draw_line`: 向任务中添加绘制线条 (含直线、斜线) 操作。
- `cvi_tde_quick_copy`: 向任务中添加快速拷贝操作。
- `cvi_tde_quick_resize`: 向任务中添加光栅位图缩放操作。
- `cvi_tde_solid_draw`: 向任务中添加对光栅位图进行有附加操作的填充搬移操作。
- `cvi_tde_rotate`: 向任务中添加对光栅位图的旋转操作。
- `cvi_tde_bit_blit`: 向任务中添加对光栅位图进行有附加功能的搬移操作。
- `cvi_tde_pattern_fill`: 模式填充。
- `cvi_tde_mb_blit`: 向任务中添加对宏块位图进行有附加功能的搬移操作。
- `cvi_tde_get_alpha_threshold_value`: 获取 alpha 判决阈值。
- `cvi_tde_set_alpha_threshold_value`: 设置 alpha 判决阈值。
- `cvi_tde_get_alpha_threshold_state`: 获取 alpha 判决开关状态。
- `cvi_tde_set_alpha_threshold_state`: 设置 alpha 判决开关状态。

3.2 功能函数参考

3.2.1 cvi_tde_open

【目的】

打开 TDE 设备。

【语法】

```
CVI_S32 cvi_tde_open(CVI_VOID);
```

【描述】

调用此接口打开 TDE 设备。

【参数】

无

【返回值】

返回值	描述
CVI_SUCCESS	成功。
错误码	失败，其值是错误码。

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

- 在进行 TDE 相关操作前应该首先调用此接口，保证 TDE 设备处于打开状态。
- TDE 设备允许多进程重复打开。

【举例】

```
/*declaration*/  
td_s32 ret = 0;  
/* open TDE device*/  
ret = cvi_tde_open ();  
if (ret != TD_SUCCESS) {  
    return -1;  
}  
/* close TDE device*/  
cvi_tde_close ();
```


3.2.2 cvi_tde_close

【目的】

关闭 TDE 设备, 将释放所有 `cvi_tde_open` 函数申请的资源。

【语法】

```
CVI_S32 cvi_tde_close(CVI_VOID);
```

【描述】

调用此接口关闭 TDE 设备。

【参数】

无

【返回值】

无

【需求】

需求	描述
头文件	<code>cvi_tde.h</code>
库文件	<code>libcvi_tde.a</code>

【注意】

调用 `cvi_tde_open` 与 `cvi_tde_close` 的次数需要对应。

【举例】

无

3.2.3 cvi_tde_begin_job

【目的】

创建 1 个 TDE 任务。

【语法】

```
CVI_S32 cvi_tde_begin_job(CVI_VOID);
```

【描述】

调用此接口创建 1 个 TDE 任务 (Job)。TDE 以任务的形式管理 TDE 命令: 1 个 TDE 任务是一系列 TDE 命令的集合, 它可以包含 1 个或多个 TDE 操作; 一个 TDE 命令对应一个 TDE 操作; 成功创建 TDE 任务添加完 TDE 操作后, 通过 `cvi_tde_end_job` 提交该 Job; 同一任务中的 TDE 命令是顺序执行。

【参数】

无

【返回值】

返回值	描述
句柄	成功
错误码	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

- 在调用此接口前应确保 TDE 设备处于打开状态。
- 应判断返回值，确保获得 1 个正确的任务句柄。
- TDE 能够缓存的任务数由 TDE 的内存大小决定，当内存不够时会出现分配任务失败的情况，建议任务数最多不要超过 200。
- cvi_tde_begin_job 必须和 cvi_tde_end_job 配套使用，否则会造成内存泄漏。

【举例】

```

/* declaration */
td_s32 ret;
td_s32 handle;
/* create a TDE job */
handle = cvi_tde_begin_job ();
if(handle == CVI_ERR_TDE_INVALID_HANDLE || handle == CVI_ERR_TDE_DEV_NOT_
↪OPEN) {
return -1;
}
/* submit the job */
ret = cvi_tde_end_job (handle, TD_FALSE, TD_TRUE, 20);
if(ret != TD_SUCCESS) {
return -1;
}

```

3.2.4 cvi_tde_end_job

【目的】

提交已创建的 TDE 任务。

【语法】

```

CVI_S32 cvi_tde_end_job(CVI_S32 handle, CVI_BOOL is_sync, CVI_BOOL is_block, CVI_U32_
↪time_out);

```

【描述】

调用此接口提交该 Job；

同一任务中的 TDE 命令是顺序执行。

- 阻塞

指该函数调用不会立刻返回，只有在以下情况下才会返回：

- TDE 任务中的命令都完成
- 等待超时
- 等待被打断

- 非阻塞

指该函数调用会立刻返回，而不关心 TDE 任务中的命令是否已经完成。

阻塞时可以设置一个最长等待时间，如果等待时间到了，TDE 任务中的命令还没有完成，函数就会提前返回，但是任务中的命令还是会在未来的某个时刻完成。

【参数】

参数名称	描述	输入/输出
handle	TDE 任务句柄。	输入
is_sync	保留参数，暂不使用。	输入
is_block	阻塞标志。TD_TRUE：阻塞，TD_FALSE：非阻塞。	输入
time_out	超时时间，单位：ms。	输入

【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

- 在调用此接口前应保证调用 cvi_tde_open 打开 TDE 设备，并且调用 cvi_tde_begin_job 获得了有效的任务句柄。
- 若设置为阻塞操作，函数超时返回或被中断返回时应该注意：此时 TDE 操作的 API 函数提前返回，但执行的操作仍会完成。
- 提交任务后，此任务对应的 handle 会变为无效，再次提交会出现错误码 CVI_ERR_TDE_INVALID_HANDLE。

【举例】

无

3.2.5 cvi_tde_quick_fill

【目的】

向任务中添加快速填充操作。

【语法】

```
CVI_S32 cvi_tde_quick_fill(CVI_S32 handle, const cvi_tde_none_src *none_src, CVI_U32 fill_
→data);
```

【描述】

将数据值 fill_data 填充到 none_src 中以 dst_surface 为目的地址、dst_rect 为输出区域的内存中，可实现颜色填充的功能。

【参数】

参数名称	描述	输入/输出
handle	TDE 任务句柄。	输入
none_src	0 源操作集合。	输入
fill_data	填充值。	输入

【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

- 支持的格式（目标位图格式）如下：

```
CVI_TDE_COLOR_FORMAT_RGB444,
CVI_TDE_COLOR_FORMAT_BGR444,
CVI_TDE_COLOR_FORMAT_RGB555,
CVI_TDE_COLOR_FORMAT_BGR555,
CVI_TDE_COLOR_FORMAT_RGB565,
CVI_TDE_COLOR_FORMAT_BGR565,
CVI_TDE_COLOR_FORMAT_RGB888,
CVI_TDE_COLOR_FORMAT_BGR888,
CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ABGR4444,
CVI_TDE_COLOR_FORMAT_RGBA4444,
CVI_TDE_COLOR_FORMAT_BGRA4444,
```

```

CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ABGR1555,
CVI_TDE_COLOR_FORMAT_RGBA5551,
CVI_TDE_COLOR_FORMAT_BGRA5551,
CVI_TDE_COLOR_FORMAT_ARGB8565,
CVI_TDE_COLOR_FORMAT_ABGR8565,
CVI_TDE_COLOR_FORMAT_RGBA8565,
CVI_TDE_COLOR_FORMAT_BGRA8565,
CVI_TDE_COLOR_FORMAT_ARGB8888,
CVI_TDE_COLOR_FORMAT_ABGR8888,
CVI_TDE_COLOR_FORMAT_RGBA8888,
CVI_TDE_COLOR_FORMAT_BGRA8888,
CVI_TDE_COLOR_FORMAT_RABG8888,
CVI_TDE_COLOR_FORMAT_CLUT1,
CVI_TDE_COLOR_FORMAT_CLUT2,
CVI_TDE_COLOR_FORMAT_CLUT4,
CVI_TDE_COLOR_FORMAT_CLUT8,
CVI_TDE_COLOR_FORMAT_ACLUT44,
CVI_TDE_COLOR_FORMAT_ACLUT88,
CVI_TDE_COLOR_FORMAT_A1,
CVI_TDE_COLOR_FORMAT_A8,
CVI_TDE_COLOR_FORMAT_YCBCR422,
CVI_TDE_COLOR_FORMAT_PKGYVYU,
CVI_TDE_COLOR_FORMAT_PKGUYVY,
CVI_TDE_COLOR_FORMAT_PKGVYUY,
CVI_TDE_COLOR_FORMAT_PKGVUYY,
CVI_TDE_COLOR_FORMAT_PKGYUYV,
CVI_TDE_COLOR_FORMAT_PKGVVYY,
CVI_TDE_COLOR_FORMAT_PKGYVVU

```

- 由于该操作直接将 fill_data 填充在位图的指定区域内，调用者欲填充蓝色到指定位图，应按照位图格式指定相应的蓝色填充值。
- 如位图格式为 ARGB1555，欲填充为蓝色，则应指定 fill_data 为 0x801F（其中 alpha 位为 1）。

【举例】

无

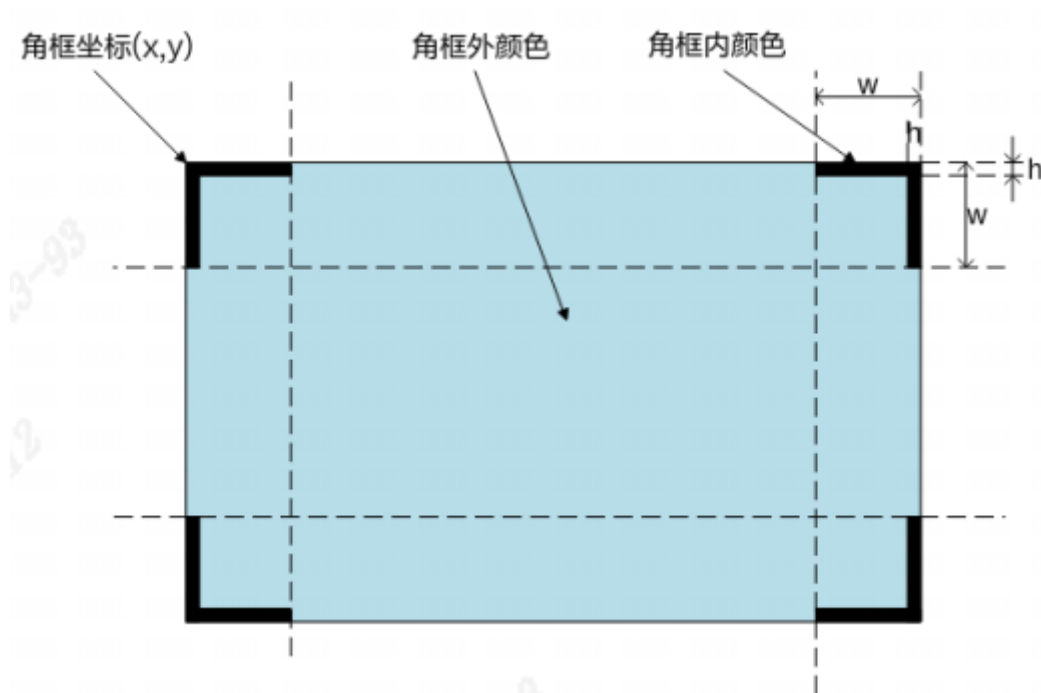
3.2.6 cvi_tde_draw_corner_box

【目的】

向任务中添加绘制复数角框操作。

【描述】

填充好结构体 corner_rect 与角框个数 num，将之绘制到以 dst_surface 为目的地址；corner_rect_region 为输出区域的内存中，可实现快速绘制复数个角框。具体如图所示：



· 说明

角框坐标 (x,y) 及角框大小 (w,h):

- x = corner_rect->corner_rect_region->pos_x;
- y = corner_rect->corner_rect_region->pos_y;
- w = corner_rect->corner_rect_region->width;
- h = corner_rect->corner_rect_region->height;

角框外颜色: corner_rect-> corner_rect_info->outer_color

角框内颜色: corner_rect-> corner_rect_info->inner_color

角框形状由 w 与 h 决定: corner_rect-> corner_rect_info -> width,
corner_rect->corner_rect_info->height

【语法】

```
CVI_S32 cvi_tde_draw_corner_box(CVI_S32 handle, const cvi_tde_surface *dst_surface, cvi_tde_
↪corner_rect *corner_rect, CVI_U32 num);
```

【参数】

参数名称	描述	输入/输出
handle	TDE 任务句柄。	输入
dst_surface	目标 surface, 参考数据结构 cvi_tde_surface。	输入
corner_rect	角框属性及位置, 参考数据结构 cvi_tde_corner_rect	输入
num	角框个数	输入

【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败, 其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

- 支持的格式（目标位图格式）如下

CVI_TDE_COLOR_FORMAT_RGB444,
 CVI_TDE_COLOR_FORMAT_BGR444,
 CVI_TDE_COLOR_FORMAT_RGB555,
 CVI_TDE_COLOR_FORMAT_BGR555,
 CVI_TDE_COLOR_FORMAT_RGB565,
 CVI_TDE_COLOR_FORMAT_BGR565,
 CVI_TDE_COLOR_FORMAT_RGB888,
 CVI_TDE_COLOR_FORMAT_BGR888,
 CVI_TDE_COLOR_FORMAT_ARGB4444,
 CVI_TDE_COLOR_FORMAT_ABGR4444,
 CVI_TDE_COLOR_FORMAT_RGBA4444,
 CVI_TDE_COLOR_FORMAT_BGRA4444,
 CVI_TDE_COLOR_FORMAT_ARGB1555,
 CVI_TDE_COLOR_FORMAT_ABGR1555,
 CVI_TDE_COLOR_FORMAT_RGBA5551,
 CVI_TDE_COLOR_FORMAT_BGRA5551,
 CVI_TDE_COLOR_FORMAT_ARGB8565,
 CVI_TDE_COLOR_FORMAT_ABGR8565,
 CVI_TDE_COLOR_FORMAT_RGBA8565,
 CVI_TDE_COLOR_FORMAT_BGRA8565,

- 使用限制

最多一次绘制 128 个框。

corner_rect_info->width 和 height 大于目标区域宽和高会自动调整为目标区域最大宽和高。

corner_rect_region->pos_x 加上 corner_rect_region->width 大于 surface.width 会重新计算

- corner_rect_region->width=dst_surface.width-corner_rect_region->pos_x

corner_rect_region->pos_y 加上 corner_rect_region->height 大于 surface.height 会重新计算

- corner_rect_region->height=dst_surface.height-corner_rect_region->pos_y

如果配置 128 个框, 当第 10 个框配置有误, 则会停止配置返回对应错误码。此时客户有以下两种处理方式:

- 通过 cvi_tde_cancel_job 取消该任务, 重新正确配置。
- 直接通过 cvi_tde_end_job 提交该任务, 正确绘制前 9 个框。

【举例】

一次绘制 64 个代码片段

```
CVI_U32 handle;
cvi_tde_corner_rect_multi_corner = {0};
/* 分配64个框的属性值的空间 */
multi_corner.corner_rect_info = (cvi_tde_corner_rect_info*)
    malloc(sizeof(cvi_tde_corner_rect_info) * 64);
multi_corner.corner_rect_region = (cvi_tde_rect*)malloc(sizeof(cvi_tde_rect) * 64);
/* 配置需要画的框与位置 */
for (i = 1; i /< 64; i++) {
    multi_corner.corner_rect_info[i].width = 4;
    multi_corner.corner_rect_info[i].height = 1;
    multi_corner.corner_rect_info[i].inner_color = 0x1;
    multi_corner.corner_rect_info[i].outer_color = 0x3;
    multi_corner.corner_rect_region[i].height = 10;
    multi_corner.corner_rect_region[i].width = 10;
    multi_corner.corner_rect_region[i].pos_x = rect_list[i - 1].pos_x + 4;
    multi_corner.corner_rect_region[i].pos_y = rect_list[i - 1].pos_y + 2;
}
/* 获取TDE有效句柄 */
handle = cvi_tde_begin_job();
/* 添加复数角框任务。dst_surface需要有效 */
cvi_tde_draw_corner_box(handle, &dst_surface, &multi_corner, 64);
/* 提交任务给硬件 */
cvi_tde_end_job(handle, is_sync, is_block, time_out);
free(multi_corner.corner_rect_info);
free(multi_corner.corner_rect_region);
cvi_tde_close();
```


3.2.7 cvi_tde_draw_line

【目的】

向任务中添加 drd 画线操作，支持画直线、斜线。

【描述】

将 num 条 line 绘制在以 dst_surface 为目的地址的内存中，可实现绘制 1 条或多条斜线（最多可绘制的线数为 100 条）；可设置的线宽为 1~256。

【语法】

```
CVI_S32 cvi_tde_draw_line(CVI_S32 handle, const cvi_tde_surface *dst_surface, const cvi_tde_
↪line *line, CVI_U32 num);
```

【参数】

参数名称	描述	输入/输出
handle	TDE 任务句柄。	输入
dst_surface	目标 surface, 参考数据结构 cvi_tde_surface。	输入
line	单一线条属性, 参考数据结构 cvi_tde_line。	输入
num	绘制线条个数。	输入

【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

- 支持的格式（目标位图格式）如下：

```
CVI_TDE_COLOR_FORMAT_RGB444,
CVI_TDE_COLOR_FORMAT_BGR444,
CVI_TDE_COLOR_FORMAT_RGB555,
CVI_TDE_COLOR_FORMAT_BGR555,
CVI_TDE_COLOR_FORMAT_RGB565,
CVI_TDE_COLOR_FORMAT_BGR565,
CVI_TDE_COLOR_FORMAT_RGB888,
CVI_TDE_COLOR_FORMAT_BGR888,
CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ABGR4444,
```

```

CVI_TDE_COLOR_FORMAT_RGBA4444,
CVI_TDE_COLOR_FORMAT_BGRA4444,
CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ABGR1555,
CVI_TDE_COLOR_FORMAT_RGBA5551,
CVI_TDE_COLOR_FORMAT_BGRA5551,
CVI_TDE_COLOR_FORMAT_ARGB8565,
CVI_TDE_COLOR_FORMAT_ABGR8565,
CVI_TDE_COLOR_FORMAT_RGBA8565,
CVI_TDE_COLOR_FORMAT_BGRA8565,
CVI_TDE_COLOR_FORMAT_ARGB8888,
CVI_TDE_COLOR_FORMAT_ABGR8888,
CVI_TDE_COLOR_FORMAT_RGBA8888,
CVI_TDE_COLOR_FORMAT_BGRA8888,
CVI_TDE_COLOR_FORMAT_RABG8888,
CVI_TDE_COLOR_FORMAT_CLUT4,
CVI_TDE_COLOR_FORMAT_CLUT8,
CVI_TDE_COLOR_FORMAT_ACLUT44,
CVI_TDE_COLOR_FORMAT_ACLUT88,
CVI_TDE_COLOR_FORMAT_A8,
CVI_TDE_COLOR_FORMAT_YCbCr888,
CVI_TDE_COLOR_FORMAT_AYCbCr8888

```

【举例】

无

3.2.8 cvi_tde_quick_copy

【目的】

向指定任务中添加快速拷贝操作。

【描述】

将 single_src 中基地址为 src_surface 的位图的指定区域 src_rect 拷贝到以 dst_surface 为目的地址、dst_rect 为输出区域的内存中。

位图、操作区域及两者的关系描述如下：

- 位图信息由 cvi_tde_surface 表示，它描述位图的基本信息，包括：位图的像素宽度、像素高度、位图每行的跨度、颜色格式、位图存放的物理地址等。
- 操作区域由 cvi_tde_rect 表示，它描述位图中参与本次操作的矩形范围，包括：起始位置和尺寸信息。

【语法】

```
CVI_S32 cvi_tde_quick_copy(CVI_S32 handle, const cvi_tde_single_src *single_src);
```

【参数】

参数名称	描述	输入/输出
handle	TDE 任务句柄。	输入
single_src	单源操作集合。	输入

【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

- 支持的格式如下：

源位图格式：

CVI_TDE_COLOR_FORMAT_RGB444,
 CVI_TDE_COLOR_FORMAT_BGR444,
 CVI_TDE_COLOR_FORMAT_RGB555,
 CVI_TDE_COLOR_FORMAT_BGR555,
 CVI_TDE_COLOR_FORMAT_RGB565,
 CVI_TDE_COLOR_FORMAT_BGR565,
 CVI_TDE_COLOR_FORMAT_RGB888,
 CVI_TDE_COLOR_FORMAT_BGR888,
 CVI_TDE_COLOR_FORMAT_ARGB4444,
 CVI_TDE_COLOR_FORMAT_ABGR4444,
 CVI_TDE_COLOR_FORMAT_RGBA4444,
 CVI_TDE_COLOR_FORMAT_BGRA4444,
 CVI_TDE_COLOR_FORMAT_ARGB1555,
 CVI_TDE_COLOR_FORMAT_ABGR1555,
 CVI_TDE_COLOR_FORMAT_RGBA5551,
 CVI_TDE_COLOR_FORMAT_BGRA5551,
 CVI_TDE_COLOR_FORMAT_ARGB8565,
 CVI_TDE_COLOR_FORMAT_ABGR8565,
 CVI_TDE_COLOR_FORMAT_RGBA8565,
 CVI_TDE_COLOR_FORMAT_BGRA8565,
 CVI_TDE_COLOR_FORMAT_ARGB8888,
 CVI_TDE_COLOR_FORMAT_ABGR8888,
 CVI_TDE_COLOR_FORMAT_RGBA8888,
 CVI_TDE_COLOR_FORMAT_BGRA8888,
 CVI_TDE_COLOR_FORMAT_RABG8888,
 CVI_TDE_COLOR_FORMAT_CLUT1,

CVI_TDE_COLOR_FORMAT_CLUT2,
CVI_TDE_COLOR_FORMAT_CLUT4,
CVI_TDE_COLOR_FORMAT_CLUT8,
CVI_TDE_COLOR_FORMAT_ACLUT44,
CVI_TDE_COLOR_FORMAT_ACLUT88,
CVI_TDE_COLOR_FORMAT_A1,
CVI_TDE_COLOR_FORMAT_A8,
CVI_TDE_COLOR_FORMAT_YCbCr888,
CVI_TDE_COLOR_FORMAT_AYCbCr888

目标位图格式:

CVI_TDE_COLOR_FORMAT_RGB444,
CVI_TDE_COLOR_FORMAT_BGR444,
CVI_TDE_COLOR_FORMAT_RGB555,
CVI_TDE_COLOR_FORMAT_BGR555,
CVI_TDE_COLOR_FORMAT_RGB565,
CVI_TDE_COLOR_FORMAT_BGR565,
CVI_TDE_COLOR_FORMAT_RGB888,
CVI_TDE_COLOR_FORMAT_BGR888,
CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ABGR4444,
CVI_TDE_COLOR_FORMAT_RGBA4444,
CVI_TDE_COLOR_FORMAT_BGRA4444,
CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ABGR1555,
CVI_TDE_COLOR_FORMAT_RGBA5551,
CVI_TDE_COLOR_FORMAT_BGRA5551,
CVI_TDE_COLOR_FORMAT_ARGB8565,
CVI_TDE_COLOR_FORMAT_ABGR8565,
CVI_TDE_COLOR_FORMAT_RGBA8565,
CVI_TDE_COLOR_FORMAT_BGRA8565,
CVI_TDE_COLOR_FORMAT_ARGB8888,
CVI_TDE_COLOR_FORMAT_ABGR8888,
CVI_TDE_COLOR_FORMAT_RGBA8888,
CVI_TDE_COLOR_FORMAT_BGRA8888,
CVI_TDE_COLOR_FORMAT_CLUT2,
CVI_TDE_COLOR_FORMAT_CLUT4,
CVI_TDE_COLOR_FORMAT_CLUT8,
CVI_TDE_COLOR_FORMAT_ACLUT44,
CVI_TDE_COLOR_FORMAT_ACLUT88,
CVI_TDE_COLOR_FORMAT_A1,
CVI_TDE_COLOR_FORMAT_A8,
CVI_TDE_COLOR_FORMAT_YCbCr888,
CVI_TDE_COLOR_FORMAT_AYCbCr888

- 此接口使用的是直接 DMA 搬移, 因此性能优于 `cvi_tde_bit_blit` 搬移。
- 快速拷贝不支持缩放功能, 因此如果源和目的的操作区域尺寸不一致, 则按照者最小的公共区域进行拷贝搬移。

- 指定的操作区域要和指定的位图有公共区域，否则会返回错误；其他操作均有此要求。

【举例】

无

3.2.9 cvi_tde_quick_resize

【目的】

向指定任务中添加快速缩放操作。

【描述】

将 single_src 中基地址为 src_surface 的位图以区域 src_rect 指定的尺寸缩放至 dst_rect 的尺寸，将结果拷贝到以 dst_surface 为目的地址、dst_rect 为输出区域的内存中。

【语法】

```
CVI_S32 cvi_tde_quick_resize(CVI_S32 handle, const cvi_tde_single_src *single_src);
```

【参数】

参数名称	描述	输入/输出
handle	TDE 任务句柄。	输入
single_src	单源操作集合。	输入

【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

- 支持的格式如下：

源位图格式：

```
CVI_TDE_COLOR_FORMAT_RGB444,
CVI_TDE_COLOR_FORMAT_BGR444,
CVI_TDE_COLOR_FORMAT_RGB555,
CVI_TDE_COLOR_FORMAT_BGR555,
CVI_TDE_COLOR_FORMAT_RGB565,
CVI_TDE_COLOR_FORMAT_BGR565,
```

CVI_TDE_COLOR_FORMAT_RGB888,
CVI_TDE_COLOR_FORMAT_BGR888,
CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ABGR4444,
CVI_TDE_COLOR_FORMAT_RGBA4444,
CVI_TDE_COLOR_FORMAT_BGRA4444,
CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ABGR1555,
CVI_TDE_COLOR_FORMAT_RGBA5551,
CVI_TDE_COLOR_FORMAT_BGRA5551,
CVI_TDE_COLOR_FORMAT_ARGB8565,
CVI_TDE_COLOR_FORMAT_ABGR8565,
CVI_TDE_COLOR_FORMAT_RGBA8565,
CVI_TDE_COLOR_FORMAT_BGRA8565,
CVI_TDE_COLOR_FORMAT_ARGB8888,
CVI_TDE_COLOR_FORMAT_ABGR8888,
CVI_TDE_COLOR_FORMAT_RGBA8888,
CVI_TDE_COLOR_FORMAT_BGRA8888,
CVI_TDE_COLOR_FORMAT_RABG8888

目标位图格式:

CVI_TDE_COLOR_FORMAT_RGB444,
CVI_TDE_COLOR_FORMAT_BGR444,
CVI_TDE_COLOR_FORMAT_RGB555,
CVI_TDE_COLOR_FORMAT_BGR555,
CVI_TDE_COLOR_FORMAT_RGB565,
CVI_TDE_COLOR_FORMAT_BGR565,
CVI_TDE_COLOR_FORMAT_RGB888,
CVI_TDE_COLOR_FORMAT_BGR888,
CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ABGR4444,
CVI_TDE_COLOR_FORMAT_RGBA4444,
CVI_TDE_COLOR_FORMAT_BGRA4444,
CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ABGR1555,
CVI_TDE_COLOR_FORMAT_RGBA5551,
CVI_TDE_COLOR_FORMAT_BGRA5551,
CVI_TDE_COLOR_FORMAT_ARGB8565,
CVI_TDE_COLOR_FORMAT_ABGR8565,
CVI_TDE_COLOR_FORMAT_RGBA8565,
CVI_TDE_COLOR_FORMAT_BGRA8565,
CVI_TDE_COLOR_FORMAT_ARGB8888,
CVI_TDE_COLOR_FORMAT_ABGR8888,
CVI_TDE_COLOR_FORMAT_RGBA8888,
CVI_TDE_COLOR_FORMAT_BGRA8888,
CVI_TDE_COLOR_FORMAT_RABG8888

· 缩小倍数小于等于 255 倍，放大倍数则没有限制 (不超过最大分辨率)。其他接口的缩放倍

数限制与此一致，不另作说明。

- 缩放时源位图和目标位图可以为同一位图，但操作区域不能有重叠，否则效果会异常。

【举例】

无

3.2.10 cvi_tde_solid_draw

【目的】

向任务中添加对光栅位图进行有附加操作的填充搬移操作。实现在 surface 上画点、画线、色块填充或内存填充等功能

【语法】

```
CVI_S32 cvi_tde_solid_draw(CVI_S32 handle, const cvi_tde_single_src *single_src, const cvi_tde_
→fill_color *fill_color, const cvi_tde_opt *opt);
```

【描述】

该接口实现背景 surface 操作区域和填充色运算后输出到目标 surface 的操作区域。该运算可以是 alpha 叠加运算或 ROP 运算，中间可以伴随着 clip 操作

【参数】

参数名称	描述	输入/输出
handle	TDE 任务句柄。	输入
single_src	单源操作集合。	输入
fill_color	填充色结构体。	输入
opt	操作属性结构体。	输入

【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】：

- 支持的格式如下：

背景位图格式：

CVI_TDE_COLOR_FORMAT_RGB444,
CVI_TDE_COLOR_FORMAT_BGR444,

CVI_TDE_COLOR_FORMAT_RGB555,
CVI_TDE_COLOR_FORMAT_BGR555,
CVI_TDE_COLOR_FORMAT_RGB565,
CVI_TDE_COLOR_FORMAT_BGR565,
CVI_TDE_COLOR_FORMAT_RGB888,
CVI_TDE_COLOR_FORMAT_BGR888,
CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ABGR4444,
CVI_TDE_COLOR_FORMAT_RGBA4444,
CVI_TDE_COLOR_FORMAT_BGRA4444,
CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ABGR1555,
CVI_TDE_COLOR_FORMAT_RGBA5551,
CVI_TDE_COLOR_FORMAT_BGRA5551,
CVI_TDE_COLOR_FORMAT_ARGB8565,
CVI_TDE_COLOR_FORMAT_ABGR8565,
CVI_TDE_COLOR_FORMAT_RGBA8565,
CVI_TDE_COLOR_FORMAT_BGRA8565,
CVI_TDE_COLOR_FORMAT_ARGB8888,
CVI_TDE_COLOR_FORMAT_ABGR8888,
CVI_TDE_COLOR_FORMAT_RGBA8888,
CVI_TDE_COLOR_FORMAT_BGRA8888,
CVI_TDE_COLOR_FORMAT_RABG8888,
CVI_TDE_COLOR_FORMAT_YCbCr888,
CVI_TDE_COLOR_FORMAT_AYCbCr8888

目标位图格式:

CVI_TDE_COLOR_FORMAT_RGB444,
CVI_TDE_COLOR_FORMAT_BGR444,
CVI_TDE_COLOR_FORMAT_RGB555,
CVI_TDE_COLOR_FORMAT_BGR555,
CVI_TDE_COLOR_FORMAT_RGB565,
CVI_TDE_COLOR_FORMAT_BGR565,
CVI_TDE_COLOR_FORMAT_RGB888,
CVI_TDE_COLOR_FORMAT_BGR888,
CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ABGR4444,
CVI_TDE_COLOR_FORMAT_RGBA4444,
CVI_TDE_COLOR_FORMAT_BGRA4444,
CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ABGR1555,
CVI_TDE_COLOR_FORMAT_RGBA5551,
CVI_TDE_COLOR_FORMAT_BGRA5551,
CVI_TDE_COLOR_FORMAT_ARGB8565,
CVI_TDE_COLOR_FORMAT_ABGR8565,
CVI_TDE_COLOR_FORMAT_RGBA8565,
CVI_TDE_COLOR_FORMAT_BGRA8565,
CVI_TDE_COLOR_FORMAT_ARGB8888,


```

CVI_TDE_COLOR_FORMAT_ABGR8888,
CVI_TDE_COLOR_FORMAT_RGBA8888,
CVI_TDE_COLOR_FORMAT_BGRA8888,
CVI_TDE_COLOR_FORMAT_RABG8888,
CVI_TDE_COLOR_FORMAT_YCbCr888,
CVI_TDE_COLOR_FORMAT_AYCbCr8888,
CVI_TDE_COLOR_FORMAT_YCBCR422,
CVI_TDE_COLOR_FORMAT_PKGYVYU,
CVI_TDE_COLOR_FORMAT_PKGUYVY,
CVI_TDE_COLOR_FORMAT_PKGVYUY,
CVI_TDE_COLOR_FORMAT_PKGVUYY,
CVI_TDE_COLOR_FORMAT_PKGYUYU,
CVI_TDE_COLOR_FORMAT_PKGUVYY,
CVI_TDE_COLOR_FORMAT_PKGYVVU

```

- 在调用此接口前应保证调用 `cvi_tde_open`: 打开 TDE 设备, 并且调用 `cvi_tde_begin_job`: 获得了有效的任务句柄。
- 当背景位图 `src_surface` 与 `src_rect` 为 `NULL` 且操作结构 `opt` 为 `NULL` 时, 该接口可实现单纯的色彩填充功能, 即与 `cvi_tde_quick_fill`: 实现的功能一样。接口调用形式如下:

```

single_src.src_surface = NULL;
single_src.src_rect = NULL;
cvi_tde_solid_draw(handle, &single_src, &fill_color, NULL);

```

- 当背景位图 `src_surface` 与 `src_rect` 不为 `NULL` 时 (此时操作属性 `opt` 一定不能为 `NULL`), 可实现背景位图的指定区域与填充色做 alpha 叠加或 ROP 等操作, 结果输出到目的位图的指定区域。接口调用形式如下:

```

//注: single_src中的值为有效值
cvi_tde_solid_draw(handle, &single_src, &fill_color, opt);

```

- 调用者可将填充色与指定位图做 alpha 叠加、ROP、colorkey、输出结果的镜像 `mirror` 和输出结果的剪切 `clip`。
 - 当指定 ROP 操作时, ROP 操作对象填充色作为前景 S2, `single_src.src_surface` 作为背景 S1。
 - 当指定 colorkey 操作时, 只能对前景 S2 做 colorkey。
- Solid Draw 绘制矩形或者水平/垂直直线的方法是通过设置填充矩形的宽/高来完成。例如: 垂直直线就是绘制宽度为 1 像素的矩形。

【举例】

无

3.2.11 cvi_tde_rotate

【目的】

向任务中添加光栅位图旋转操作。

【语法】

```
CVI_S32 cvi_tde_rotate(td_s32 handle, const cvi_tde_single_src *single_src, cvi_tde_rotate_
↪angle rotate);
```

【描述】

将 single_src 中基地址为 src_surface 的位图以区域 src_rect 指定的尺寸旋转至 dst_rect 的尺寸，将结果拷贝到以 dst_surface 为目的地址、dst_rect 为输出区域的内存中，可以做 90 度，180 度和 270 度顺时针旋转。

【参数】

参数名称	描述	输入/输出
handle	TDE 任务句柄。	输入
single_src	单源操作集合。	输入
rotate	旋转的角度	输入

【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

- 支持的格式如下：

```
CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ARGB8888,
CVI_TDE_COLOR_FORMAT_YCbCr422,
CVI_TDE_COLOR_FORMAT_PKGYVYU,
CVI_TDE_COLOR_FORMAT_PKGUYVY,
CVI_TDE_COLOR_FORMAT_PKGVYUY,
CVI_TDE_COLOR_FORMAT_PKGVUYY,
CVI_TDE_COLOR_FORMAT_PKGYUYU,
CVI_TDE_COLOR_FORMAT_PKGVVYY,
CVI_TDE_COLOR_FORMAT_PKGYVVU
```

- 旋转的同时不支持格式转换，即输入输出像素格式需一致。
- 当像素格式是 YCbCr422 时，操作区域宽、高需要偶对齐。
- 位图物理地址与行间距需要 16 对齐。
- 旋转的同时不叠加任何其余规格。
- 输入格式为 CVI_TDE_COLOR_FORMAT_YCbCr422 时，做 180 度旋转，不能超过 zme 的行 buffer 约束，具体约束见 `cvi_tde_quick_resize` 接口中的约束描述。

【举例】

无

3.2.12 `cvi_tde_bit_blit`

【目的】

向任务中添加对光栅位图进行有附加功能的搬移操作。

【语法】

```
CVI_S32 cvi_tde_bit_blit(CVI_S32 handle, const cvi_tde_double_src *double_src, const cvi_tde_opt *opt);
```

【描述】

将 `double_src` 中前景位图 (`fg_surface`) 与背景位图 (`bg_surface`) 的指定区域 (`fg_rect`、`bg_rect`) 进行运算，将运算后的位图拷贝到目标位图 (`dst_surface`) 的指定区域 (`dst_rect`) 中。其中当前景位图不为 NULL 时，背景位图 (`bg_surface`) 的指定区域 (`bg_rect`) 和目标位图 (`dst_surface`) 的指定区域 (`dst_rect`) 必须一致。

`cvi_tde_opt` 结构中存放有 TDE 运算功能的配置信息，如：是否进行 ROP 操作及 ROP 命令码；是否作色键 (`colorkey`) 及 `colorkey` 的配置值；是否作区域裁减 (`clip` 操作) 及指定 `clip` 区域；是否缩放、是否抗闪烁、是否镜像、是否进行 `alpha` 混合等信息。上述的操作可以同时使能。

`cvi_tde_opt` 结构中的配置项涉及到的概念解释，具体参考 **【功能说明】**：

- 单源或双源的图形操作

单源操作指只有一个位图来源（如仅指定背景位图和目的位图，前景位图为 NULL），针对该位图可以做以下处理：

- * 位图搬移
- * 位图格式转换
- * 位图缩放
- * 位图抗闪烁
- * 位图颜色扩展或颜色校正
- * 位图输出结果裁减，即 `clip`

双源操作指有两个位图来源（背景位图和前景位图），两个位图的运算结果输出到目的位图指定的区域。其中，背景位图可以与目标位图为同一位图，此操作的含义为：将前景位图与背景位图进行运算，将结果直接输出覆盖到背景位图中。双源类的操作包括以下处理：

- * 前景和背景的 ROP 操作

- * 前景和背景的 alpha 叠加操作
- * ColorKey 操作
- * 前景位图指定区域缩放/抗闪烁处理后，再与背景做 alpha 叠加等操作

【参数】

参数名称	描述	输入/输出
handle	TDE 任务句柄。	输入
double_src	双源操作集合。	输入
opt	运算参数设置结构。	输入

【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

- 支持的格式如下：

背景位图格式：

```

CVI_TDE_COLOR_FORMAT_RGB444,
CVI_TDE_COLOR_FORMAT_BGR444,
CVI_TDE_COLOR_FORMAT_RGB555,
CVI_TDE_COLOR_FORMAT_BGR555,
CVI_TDE_COLOR_FORMAT_RGB565,
CVI_TDE_COLOR_FORMAT_BGR565,
CVI_TDE_COLOR_FORMAT_RGB888,
CVI_TDE_COLOR_FORMAT_BGR888,
CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ABGR4444,
CVI_TDE_COLOR_FORMAT_RGBA4444,
CVI_TDE_COLOR_FORMAT_BGRA4444,
CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ABGR1555,
CVI_TDE_COLOR_FORMAT_RGBA5551,
CVI_TDE_COLOR_FORMAT_BGRA5551,
CVI_TDE_COLOR_FORMAT_ARGB8565,
CVI_TDE_COLOR_FORMAT_ABGR8565,
CVI_TDE_COLOR_FORMAT_RGBA8565,
CVI_TDE_COLOR_FORMAT_BGRA8565,

```

CVI_TDE_COLOR_FORMAT_ARGB8888,
CVI_TDE_COLOR_FORMAT_ABGR8888,
CVI_TDE_COLOR_FORMAT_RGBA8888,
CVI_TDE_COLOR_FORMAT_BGRA8888,
CVI_TDE_COLOR_FORMAT_RABG8888,
CVI_TDE_COLOR_FORMAT_YCbCr888,
CVI_TDE_COLOR_FORMAT_RABG8888,
CVI_TDE_COLOR_FORMAT_CLUT1,
CVI_TDE_COLOR_FORMAT_CLUT2,
CVI_TDE_COLOR_FORMAT_CLUT4,
CVI_TDE_COLOR_FORMAT_CLUT8,
CVI_TDE_COLOR_FORMAT_ACLUT44,
CVI_TDE_COLOR_FORMAT_ACLUT88,
CVI_TDE_COLOR_FORMAT_A1,
CVI_TDE_COLOR_FORMAT_A8,
CVI_TDE_COLOR_FORMAT_YCbCr888,
CVI_TDE_COLOR_FORMAT_AYCbCr8888

前景位图格式:

CVI_TDE_COLOR_FORMAT_RGB444,
CVI_TDE_COLOR_FORMAT_BGR444,
CVI_TDE_COLOR_FORMAT_RGB555,
CVI_TDE_COLOR_FORMAT_BGR555,
CVI_TDE_COLOR_FORMAT_RGB565,
CVI_TDE_COLOR_FORMAT_BGR565,
CVI_TDE_COLOR_FORMAT_RGB888,
CVI_TDE_COLOR_FORMAT_BGR888,
CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ABGR4444,
CVI_TDE_COLOR_FORMAT_RGBA4444,
CVI_TDE_COLOR_FORMAT_BGRA4444,
CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ABGR1555,
CVI_TDE_COLOR_FORMAT_RGBA5551,
CVI_TDE_COLOR_FORMAT_BGRA5551,
CVI_TDE_COLOR_FORMAT_ARGB8565,
CVI_TDE_COLOR_FORMAT_ABGR8565,
CVI_TDE_COLOR_FORMAT_RGBA8565,
CVI_TDE_COLOR_FORMAT_BGRA8565,
CVI_TDE_COLOR_FORMAT_ARGB8888,
CVI_TDE_COLOR_FORMAT_ABGR8888,
CVI_TDE_COLOR_FORMAT_RGBA8888,
CVI_TDE_COLOR_FORMAT_BGRA8888,
CVI_TDE_COLOR_FORMAT_RABG8888,
CVI_TDE_COLOR_FORMAT_CLUT1,
CVI_TDE_COLOR_FORMAT_CLUT2,
CVI_TDE_COLOR_FORMAT_CLUT4,
CVI_TDE_COLOR_FORMAT_CLUT8,

CVI_TDE_COLOR_FORMAT_ACLUT44,
CVI_TDE_COLOR_FORMAT_ACLUT88,
CVI_TDE_COLOR_FORMAT_A1,
CVI_TDE_COLOR_FORMAT_A8,
CVI_TDE_COLOR_FORMAT_YCbCr888,
CVI_TDE_COLOR_FORMAT_AYCbCr8888,
CVI_TDE_COLOR_FORMAT_YCbCr422,
CVI_TDE_COLOR_FORMAT_PKGYVYU,
CVI_TDE_COLOR_FMT_PKGUYVY
CVI_TDE_COLOR_FMT_PKGVYUY,
CVI_TDE_COLOR_FMT_PKGVUYY,
CVI_TDE_COLOR_FMT_PKGYUYV,
CVI_TDE_COLOR_FMT_PKGUVYY,
CVI_TDE_COLOR_FMT_PKGYVVU,
CVI_TDE_COLOR_FORMAT_JPG_YCbCr400MBP,
CVI_TDE_COLOR_FORMAT_JPG_YCbCr422MBHP,
CVI_TDE_COLOR_FORMAT_JPG_YCbCr422MBVP,
CVI_TDE_COLOR_FORMAT_MP1_YCbCr420MBP,
CVI_TDE_COLOR_FORMAT_MP2_YCbCr420MBP,
CVI_TDE_COLOR_FORMAT_MP2_YCbCr420MBI,
CVI_TDE_COLOR_FORMAT_JPG_YCbCr420MBP,
CVI_TDE_COLOR_FORMAT_JPG_YCbCr444MBP

目标位图格式:

CVI_TDE_COLOR_FORMAT_RGB444,
CVI_TDE_COLOR_FORMAT_BGR444,
CVI_TDE_COLOR_FORMAT_RGB555,
CVI_TDE_COLOR_FORMAT_BGR555,
CVI_TDE_COLOR_FORMAT_RGB565,
CVI_TDE_COLOR_FORMAT_BGR565,
CVI_TDE_COLOR_FORMAT_RGB888,
CVI_TDE_COLOR_FORMAT_BGR888,
CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ABGR4444,
CVI_TDE_COLOR_FORMAT_RGBA4444,
CVI_TDE_COLOR_FORMAT_BGRA4444,
CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ABGR1555,
CVI_TDE_COLOR_FORMAT_RGBA5551,
CVI_TDE_COLOR_FORMAT_BGRA5551,
CVI_TDE_COLOR_FORMAT_ARGB8565,
CVI_TDE_COLOR_FORMAT_ABGR8565,
CVI_TDE_COLOR_FORMAT_RGBA8565,
CVI_TDE_COLOR_FORMAT_BGRA8565,
CVI_TDE_COLOR_FORMAT_ARGB8888,
CVI_TDE_COLOR_FORMAT_ABGR8888,
CVI_TDE_COLOR_FORMAT_RGBA8888,

```

CVI_TDE_COLOR_FORMAT_BGRA8888,
CVI_TDE_COLOR_FORMAT_RABG8888,
CVI_TDE_COLOR_FORMAT_CLUT1,
CVI_TDE_COLOR_FORMAT_CLUT2,
CVI_TDE_COLOR_FORMAT_CLUT4,
CVI_TDE_COLOR_FORMAT_CLUT8,
CVI_TDE_COLOR_FORMAT_ACLUT44,
CVI_TDE_COLOR_FORMAT_ACLUT88,
CVI_TDE_COLOR_FORMAT_A1,
CVI_TDE_COLOR_FORMAT_A8,
CVI_TDE_COLOR_FORMAT_YCbCr888,CVI_TDE_C
OLOR_FORMAT_AYCbCr8888

```

- Clut 格式作为输出时，输入格式需与输出格式保持一致，且只能做拷贝操作。且不能两个源同为 Clut 格式。
- 只有单源操作且操作项为 NULL 时，支持前景和目标为 A1、A8 格式，但前景和目标的格式必须相同，其余操作均不支持 A1、A8 格式。
- 在调用此接口前应保证调用:ref:cvi_tde_open <cvi_tde_open> 打开 TDE 设备，并且调用:ref:cvi_tde_begin_job <cvi_tde_begin_job> 获得了有效的任务句柄。
- 目标位图必须与背景位图的颜色空间一致，前景位图的颜色空间可以与背景/目标位图不一致，这种情况下会进行颜色空间转换功能。
- 当前景源位图与目标位图尺寸不一致时，如果设置了缩放则按照设定的区域进行缩放，否则按照设置公共区域的最小值进行裁减搬移。
- global_alpha 和 alpha0、alpha1 的设置值统一按照 [0, 255] 的范围进行设置。
- 背景位图可以与目标位图为同一位图。
- 当只需要使用单源搬移操作时（比如只对源位图进行 ROP 取非操作），可以将背景或背景位图的结构信息和操作区域结构指针设置为空。
- clip 操作时
 - 若为区域内 clip，则裁减区域必须与操作区域有公共交集，否则会返回错误。
 - 若为区域外 clip，则裁减区域不可完全覆盖操作区域，否则会返回错误码。也就是说，实际更新区域不能为空。
- 在第一次作颜色扩展操作（源为 Clut 格式，目的为 ARGB/AYCbCr 格式）时，需要打开 clut_reload 标记。
- ROP 操作时，通过操作结构体:ref:cvi_tde_opt <cvi_tde_opt>: 中的成员 rop_color 和 rop_alpha 分别指定颜色和 alpha 分量进行的 ROP 操作。其中，ROP 操作类型中的 S1 指背景位图 bg_surface，S2 指前景位图 fg_surface。
- 伴随缩放的情况下需满足:ref:cvi_tde_quick_resize <cvi_tde_quick_resize> 对宽高的限制。

【举例】

无

3.2.13 cvi_tde_pattern_fill

【目的】

模式填充。

【语法】

```
CVI_S32 cvi_tde_pattern_fill(CVI_S32 handle, const cvi_tde_double_src *double_src, const cvi_tde_pattern_fill_opt *fill_opt);
```

【描述】：

将前景位图 (fg_surface) 的指定区域 (fg_rect) 平铺到背景位图 (bg_surface) 的指定区域 (bg_rect)，平铺的过程中可以实现 colorkey, ROP, clip, 颜色扩展, 位图格式转换等操作，将操作后的结果搬移到目标位图 (dst_surface) 的指定区域 (dst_rect)。将前景位图往背景位图模式填充时，前景位图的指定区域不会进行缩放，前景位图会平铺到整个背景位图的指定区域。若前景位图的指定区域大于背景位图的指定区域，则自动进行裁减。

- 当只需要使用单源操作时，可以将背景位图和指定区域置空或前景位图和指定区域置空，此时可以直接将前景位图或背景位图平铺到目的位图的指定区域。平铺过程可以实现位图格式转换，位图颜色扩展或颜色校正，位图输出结果裁减，即 clip。
- 当使用双源操作时，前景位图的指定区域填充到背景位图的指定区域中时，两个位图可以先做运算，将运算结果输出到目的位图指定的区域。双源的操作包括以下处理：
 - 前景和背景的 ROP 操作
 - 前景和背景的 alpha 叠加操作
 - ColorKey 操作

【参数】

参数名称	描述	输入/输出
handle	TDE 任务句柄。	输入
double_src	双源操作集合。	输入
fill_opt	运算参数设置结构。	输入

【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

- 支持的格式如下：

背景位图格式:

CVI_TDE_COLOR_FORMAT_RGB444,
CVI_TDE_COLOR_FORMAT_BGR444,
CVI_TDE_COLOR_FORMAT_RGB555,
CVI_TDE_COLOR_FORMAT_BGR555,
CVI_TDE_COLOR_FORMAT_RGB565,
CVI_TDE_COLOR_FORMAT_BGR565,
CVI_TDE_COLOR_FORMAT_RGB888,
CVI_TDE_COLOR_FORMAT_BGR888,
CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ABGR4444,
CVI_TDE_COLOR_FORMAT_RGBA4444,
CVI_TDE_COLOR_FORMAT_BGRA4444,
CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ABGR1555,
CVI_TDE_COLOR_FORMAT_RGBA5551,
CVI_TDE_COLOR_FORMAT_BGRA5551,
CVI_TDE_COLOR_FORMAT_ARGB8565,
CVI_TDE_COLOR_FORMAT_ABGR8565,
CVI_TDE_COLOR_FORMAT_RGBA8565,
CVI_TDE_COLOR_FORMAT_BGRA8565,
CVI_TDE_COLOR_FORMAT_ARGB8888,
CVI_TDE_COLOR_FORMAT_ABGR8888,
CVI_TDE_COLOR_FORMAT_RGBA8888,
CVI_TDE_COLOR_FORMAT_BGRA8888,
CVI_TDE_COLOR_FORMAT_RABG8888,
CVI_TDE_COLOR_FORMAT_YCbCr888,
CVI_TDE_COLOR_FORMAT_RABG8888,
CVI_TDE_COLOR_FORMAT_CLUT1,
CVI_TDE_COLOR_FORMAT_CLUT2,
CVI_TDE_COLOR_FORMAT_CLUT4,
CVI_TDE_COLOR_FORMAT_CLUT8,
CVI_TDE_COLOR_FORMAT_ACLUT44,
CVI_TDE_COLOR_FORMAT_ACLUT88,
CVI_TDE_COLOR_FORMAT_A1,
CVI_TDE_COLOR_FORMAT_A8,
CVI_TDE_COLOR_FORMAT_YCbCr888,
CVI_TDE_COLOR_FORMAT_AYCbCr8888

前景位图格式:

CVI_TDE_COLOR_FORMAT_RGB444,
CVI_TDE_COLOR_FORMAT_BGR444,
CVI_TDE_COLOR_FORMAT_RGB555,
CVI_TDE_COLOR_FORMAT_BGR555,
CVI_TDE_COLOR_FORMAT_RGB565,
CVI_TDE_COLOR_FORMAT_BGR565,
CVI_TDE_COLOR_FORMAT_RGB888,
CVI_TDE_COLOR_FORMAT_BGR888,

CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ABGR4444,
CVI_TDE_COLOR_FORMAT_RGBA4444,
CVI_TDE_COLOR_FORMAT_BGRA4444,
CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ABGR1555,
CVI_TDE_COLOR_FORMAT_RGBA5551,
CVI_TDE_COLOR_FORMAT_BGRA5551,
CVI_TDE_COLOR_FORMAT_ARGB8565,
CVI_TDE_COLOR_FORMAT_ABGR8565,
CVI_TDE_COLOR_FORMAT_RGBA8565,
CVI_TDE_COLOR_FORMAT_BGRA8565,
CVI_TDE_COLOR_FORMAT_ARGB8888,
CVI_TDE_COLOR_FORMAT_ABGR8888,
CVI_TDE_COLOR_FORMAT_RGBA8888,
CVI_TDE_COLOR_FORMAT_BGRA8888,
CVI_TDE_COLOR_FORMAT_RABG8888,
CVI_TDE_COLOR_FORMAT_CLUT1,
CVI_TDE_COLOR_FORMAT_CLUT2,
CVI_TDE_COLOR_FORMAT_CLUT4,
CVI_TDE_COLOR_FORMAT_CLUT8,
CVI_TDE_COLOR_FORMAT_ACLUT44,
CVI_TDE_COLOR_FORMAT_ACLUT88,
CVI_TDE_COLOR_FORMAT_A1,
CVI_TDE_COLOR_FORMAT_A8,
CVI_TDE_COLOR_FORMAT_YCbCr888,
CVI_TDE_COLOR_FORMAT_AYCbCr8888

目标位图格式:

CVI_TDE_COLOR_FORMAT_RGB444,
CVI_TDE_COLOR_FORMAT_BGR444,
CVI_TDE_COLOR_FORMAT_RGB555,
CVI_TDE_COLOR_FORMAT_BGR555,
CVI_TDE_COLOR_FORMAT_RGB565,
CVI_TDE_COLOR_FORMAT_BGR565,
CVI_TDE_COLOR_FORMAT_RGB888,
CVI_TDE_COLOR_FORMAT_BGR888,
CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ABGR4444,
CVI_TDE_COLOR_FORMAT_RGBA4444,
CVI_TDE_COLOR_FORMAT_BGRA4444,
CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ABGR1555,
CVI_TDE_COLOR_FORMAT_RGBA5551,
CVI_TDE_COLOR_FORMAT_BGRA5551,
CVI_TDE_COLOR_FORMAT_ARGB8565,
CVI_TDE_COLOR_FORMAT_ABGR8565,
CVI_TDE_COLOR_FORMAT_RGBA8565,

```
CVI_TDE_COLOR_FORMAT_BGRA8565,
CVI_TDE_COLOR_FORMAT_ARGB8888,
CVI_TDE_COLOR_FORMAT_ABGR8888,
CVI_TDE_COLOR_FORMAT_RGBA8888,
CVI_TDE_COLOR_FORMAT_BGRA8888,
CVI_TDE_COLOR_FORMAT_RABG8888,
CVI_TDE_COLOR_FORMAT_CLUT1,
CVI_TDE_COLOR_FORMAT_RABG8888,
CVI_TDE_COLOR_FORMAT_CLUT8,
CVI_TDE_COLOR_FORMAT_YcbCr888,
CVI_TDE_COLOR_FORMAT_AYCbCr8888
```

- 在调用此接口前应保证调用:ref:cvi_tde_open <cvi_tde_open>: 打开 TDE 设备, 并且调用:ref:cvi_tde_begin_job <cvi_tde_begin_job>: 获得了有效的任务句柄。
- 若背景位图为 NULL 时, 则当前景位图指定区域大于目标位图指定区域, 进行裁减。
- 若背景位图指定区域没超过背景位图最大宽高且目标位图指定区域没超过目标位图最大宽高, 则背景位图指定区域大小和目标位图指定区域大小必须一致。
- 若目标位图的指定区域大于目标位图的最大宽高, 则自动裁减; 而前景位图或背景位图的宽高超过其对应位图的最大宽高, 则不会进行裁减, 格式填充不成功。
- 若前景位图的指定区域大于背景位图的指定区域, 则自动裁减。
- 若背景与前景的像素格式不一致, 则目的位图可以为除 clut 格式外的其他格式。且背景位图和目标位图的颜色空间可以不一致。
- 若前景位图和背景位图均不为 NULL, 则在将前景位图的指定区域往背景位图的指定区域填充的过程中不可做缩放, 抗闪和镜像操作, 其他操作和 Bitblit 中两幅位图可做的操作相同。
- 作 clip 操作时, 裁减区域必须与操作区域有公共交集, 否则会返回错误。
- 在第一次作颜色扩展操作 (源为 Clut 格式, 目的为 ARGB/AYCbCr 格式), 需要打开 Clut Reload 标记。
- ROP 操作时, 通过操作结构体 cvi_tde_opt 中的成员 rop_color 和 rop_alpha 分别指定颜色和 alpha 分量进行的 ROP 操作。其中, ROP 操作类型中的 S1 指背景位图 bg_surface, S2 指前景位图 fg_surface。
- 该操作不能做缩放。
- 当单源操作时, 无论是背景或是前景, 其指定区域的宽不可超过 256; 当双源操作时, 前景指定区域的宽不可超过 256。

【举例】

无

3.2.14 cvi_tde_mb_blit

【目的】

向任务中添加对宏块位图进行有附加功能的搬移操作。将亮度和色度宏块数据合并成光栅格式，可以伴随缩放、抗闪烁、clip 处理。

【语法】

```
CVI_S32 cvi_tde_mb_blit(CVI_S32 handle, const cvi_tde_mb_src *mb_src, const cvi_tde_mb_
    ↪opt *opt);
```

【描述】：

宏块 surface 指定区域的亮度和色度数据合并成光栅格式输出到目标 surface 的指定区域。在合并的过程中可以伴随缩放操作，由 opt 的 resize_en 参数指定。如果没有指定缩放，将直接将宏块数据合并的结果输出到目标 surface 上，超出的部分将剪切掉。当 clip 开关打开时，将做剪切拷贝；合并过程中也支持抗闪烁处理。

【参数】

参数名称	描述	输入/输出
handle	TDE 任务句柄。	输入
mb_src	宏块操作集合。	输入
opt	宏块操作属性。	输入

【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

- 支持的格式如下：

前景位图格式：

```
CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr400MBP,
CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr422MBHP,
CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr422MBVP,
CVI_TDE_MB_COLOR_FORMAT_MP1_YCbCr420MBP,
CVI_TDE_MB_COLOR_FORMAT_MP2_YCbCr420MBP,
CVI_TDE_MB_COLOR_FORMAT_MP2_YCbCr420MBI,
CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr420MBP,
```

CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr444MBP

目标位图格式:

CVI_TDE_COLOR_FORMAT_RGB444,
CVI_TDE_COLOR_FORMAT_BGR444,
CVI_TDE_COLOR_FORMAT_RGB555,
CVI_TDE_COLOR_FORMAT_BGR555,
CVI_TDE_COLOR_FORMAT_RGB565,
CVI_TDE_COLOR_FORMAT_BGR565,
CVI_TDE_COLOR_FORMAT_RGB888,
CVI_TDE_COLOR_FORMAT_BGR888,
CVI_TDE_COLOR_FORMAT_ARGB4444,
CVI_TDE_COLOR_FORMAT_ABGR4444,
CVI_TDE_COLOR_FORMAT_RGBA4444,
CVI_TDE_COLOR_FORMAT_BGRA4444,
CVI_TDE_COLOR_FORMAT_ARGB1555,
CVI_TDE_COLOR_FORMAT_ABGR1555,
CVI_TDE_COLOR_FORMAT_RGBA5551,
CVI_TDE_COLOR_FORMAT_BGRA5551,
CVI_TDE_COLOR_FORMAT_ARGB8565,
CVI_TDE_COLOR_FORMAT_ABGR8565,
CVI_TDE_COLOR_FORMAT_RGBA8565,
CVI_TDE_COLOR_FORMAT_BGRA8565,
CVI_TDE_COLOR_FORMAT_ARGB8888,
CVI_TDE_COLOR_FORMAT_ABGR8888,
CVI_TDE_COLOR_FORMAT_RGBA8888,
CVI_TDE_COLOR_FORMAT_BGRA8888,
CVI_TDE_COLOR_FORMAT_RABG8888,
CVI_TDE_COLOR_FORMAT_YcbCr888,
CVI_TDE_COLOR_FORMAT_AYCbCr8888,
CVI_TDE_COLOR_FORMAT_YCbCr422,
CVI_TDE_COLOR_FORMAT_PKGYVYU,
CVI_TDE_COLOR_FORMAT_PKGUYVY,
CVI_TDE_COLOR_FORMAT_PKGVYUY,
CVI_TDE_COLOR_FORMAT_PKGVUYU,
CVI_TDE_COLOR_FORMAT_PKGYUYU,
CVI_TDE_COLOR_FORMAT_PKGUVYU,
CVI_TDE_COLOR_FORMAT_PKGYUVU

- 在调用此接口前应保证调用`cvi_tde_open`: 打开 TDE 设备, 并且调用`cvi_tde_begin_job`: 获得了有效的任务句柄。
- 对于 YCbCr422 格式的宏块, 若采用的是水平方向采样, 则操作区域起始点水平坐标必须是偶数。若垂直方向采样, 无此限制。
- 在调用此接口前应保证调用`cvi_tde_open`: 打开 TDE 设备, 并且调用`cvi_tde_begin_job`: 获得了有效的任务句柄。

【举例】

无

3.2.15 cvi_tde_get_alpha_threshold_value

【目的】

获取 alpha 判决阈值。

【语法】

```
CVI_S32 cvi_tde_get_alpha_threshold_value(CVI_U8 *threshold_value);
```

【描述】

获取 alpha 判决阈值。用于结果图片像素格式为 ARGB1555 的情况。若前景位图和背景位图的 alpha 运算结果小于此阈值，结果像素的 alpha 位取 0；大于或等于此阈值，像素的 alpha 位取 1。

【参数】

参数名称	描述	输入/输出
threshold_value	指向 alpha 判决阈值的指针	输出

【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

无

【举例】

无

3.2.16 cvi_tde_set_alpha_threshold_value

【目的】

设置 alpha 判决阈值。

【语法】

```
CVI_S32 cvi_tde_set_alpha_threshold_value(CVI_U8 threshold_value);
```

【描述】

设置 alpha 判决阈值。当前景和背景做 bitblit 操作时，不管前景位图和背景位图的格式是什么，硬件都会生成 ARGB8888 的中间位图格式，若目标图片像素格式为 ARGB1555 的情况，则前景位图和背景位图的 alpha 运算结果小于此阈值，结果像素的 alpha 位取 0；大于或等于此阈值，像素的 alpha 位取 1。

【参数】

参数名称	描述	输入/输出
threshold_value	alpha 判决阈值	输入

【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

无

【举例】

无

3.2.17 cvi_tde_get_alpha_threshold_state

【目的】

获取 alpha 判决开关状态。

【语法】

```
CVI_S32 cvi_tde_get_alpha_threshold_state(CVI_BOOL *threshold_en);
```

【描述】

获取 alpha 判决开关状态。

【参数】

参数名称	描述	输入/输出
threshold_en	指向 alpha 判决开关状态的指针。	输出

【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

无

【举例】

无

3.2.18 cvi_tde_set_alpha_threshold_state

【目的】

设置 alpha 判决开关状态。

【语法】

```
CVI_S32 cvi_tde_set_alpha_threshold_state(CVI_BOOL threshold_en);
```

【描述】

设置 alpha 判决开关状态。在开关开的状态下，阈值为用户自己设置的值，否则阈值为 0xFF。

【参数】

参数名称	描述	输入/输出
threshold_en	alpha 判决开关状态。FALSE：判决开关关。 TRUE：判决开关开；	输出

【返回值】

返回值	描述
0	成功
非 0	失败，其值是错误码

【需求】

需求	描述
头文件	cvi_tde.h
库文件	libcvi_tde.a

【注意】

无

【举例】

无

4 数据类型

本章列举并描述可配置的数据结构。

4.1 映射表

表 4.1: 数据类型映射表

数据类型	说明
cvi_tde_color_format	TDE 支持的像素格式枚举
cvi_tde_mb_color_format	TDE 支持的宏块像素格式
cvi_tde_surface	位图 surface 结构体
cvi_tde_mb_surface	宏块格式位图基本属性
cvi_tde_rect	操作区域属性
cvi_tde_none_src	无源操作结构体
cvi_tde_single_src	单源操作结构体
cvi_tde_mb_src	宏块源操作结构体
cvi_tde_double_src	双源操作结构体
cvi_tde_alpha_blending	TDE 逻辑运算类型
cvi_tde_rop_mode	TDE 支持的 ROP 操作类型
cvi_tde_mirror_mode	图像镜像属性
cvi_tde_clip_mode	剪切操作类型
cvi_tde_mb_resize	宏块格式缩放类型
cvi_tde_fill_color	图像填充色属性结构
cvi_tde_colorkey_mode	Colorkey 模式
cvi_tde_colorkey_component	单个颜色分量的关键色属性
cvi_tde_colorkey	Colorkey 关键色属性
cvi_tde_out_alpha_from	输出 alpha 来源类型
cvi_tde_filter_mode	图像滤波模式
cvi_tde_blend_mode	用户自定义 alpha 混合模式
cvi_tde_blend_cmd	Alpha 混合命令
cvi_tde_blend_opt	alpha 混合操作选项
cvi_tde_csc_opt	CSC 参数选项
cvi_tde_opt	TDE 操作属性结构体
cvi_tde_mb_opt	宏块 Surface 的操作属性
cvi_tde_pattern_fill_opt	模式填充操作信息定义

下页继续

表 4.1 – 续上页

cvi_tde_rotate_angle	旋转角度
cvi_tde_corner_rect_info	角框属性结构体
cvi_tde_corner_rect	角框属性与角框绘制位置的定义
cvi_tde_line	Drd 画线功能线条属性的定义。

4.2 详细描述

4.2.1 cvi_tde_color_format

【说明】

TDE 支持的像素格式枚举。

【定义】

```
typedef enum {  
    CVI_TDE_COLOR_FORMAT_RGB444 = 0,  
    CVI_TDE_COLOR_FORMAT_BGR444,  
    CVI_TDE_COLOR_FORMAT_RGB555,  
    CVI_TDE_COLOR_FORMAT_BGR555,  
    CVI_TDE_COLOR_FORMAT_RGB565,  
    CVI_TDE_COLOR_FORMAT_BGR565,  
    CVI_TDE_COLOR_FORMAT_RGB888,  
    CVI_TDE_COLOR_FORMAT_BGR888,  
    CVI_TDE_COLOR_FORMAT_ARGB4444,  
    CVI_TDE_COLOR_FORMAT_ABGR4444,  
    CVI_TDE_COLOR_FORMAT_RGBA4444,  
    CVI_TDE_COLOR_FORMAT_BGRA4444,  
    CVI_TDE_COLOR_FORMAT_ARGB1555,  
    CVI_TDE_COLOR_FORMAT_ABGR1555,  
    CVI_TDE_COLOR_FORMAT_RGBA5551,  
    CVI_TDE_COLOR_FORMAT_BGRA5551,  
    CVI_TDE_COLOR_FORMAT_ARGB8565,  
    CVI_TDE_COLOR_FORMAT_ABGR8565,  
    CVI_TDE_COLOR_FORMAT_RGBA8565,  
    CVI_TDE_COLOR_FORMAT_BGRA8565,  
    CVI_TDE_COLOR_FORMAT_ARGB8888,  
    CVI_TDE_COLOR_FORMAT_ABGR8888,  
    CVI_TDE_COLOR_FORMAT_RGBA8888,  
    CVI_TDE_COLOR_FORMAT_BGRA8888,  
    CVI_TDE_COLOR_FORMAT_RABG8888,  
    CVI_TDE_COLOR_FORMAT_CLUT1,  
    CVI_TDE_COLOR_FORMAT_CLUT2,  
    CVI_TDE_COLOR_FORMAT_CLUT4,  
    CVI_TDE_COLOR_FORMAT_CLUT8,  
    CVI_TDE_COLOR_FORMAT_ACLUT44,  
    CVI_TDE_COLOR_FORMAT_ACLUT88,  
    CVI_TDE_COLOR_FORMAT_A1,  
}
```

(下页继续)

(续上页)

```

CVI_TDE_COLOR_FORMAT_A8,
CVI_TDE_COLOR_FORMAT_YCbCr888,
CVI_TDE_COLOR_FORMAT_AYCbCr8888,
CVI_TDE_COLOR_FORMAT_YCbCr422,
CVI_TDE_COLOR_FORMAT_PKGYVYU,
CVI_TDE_COLOR_FORMAT_PKGUYVY,
CVI_TDE_COLOR_FORMAT_PKGVYUY,
CVI_TDE_COLOR_FORMAT_PKGVUYY,
CVI_TDE_COLOR_FORMAT_PKGYUYU,
CVI_TDE_COLOR_FORMAT_PKGUVYY,
CVI_TDE_COLOR_FORMAT_PKGYVVU,
CVI_TDE_COLOR_FORMAT_JPG_YCbCr400MBP,
CVI_TDE_COLOR_FORMAT_JPG_YCbCr422MBHP,
CVI_TDE_COLOR_FORMAT_JPG_YCbCr422MBVP,
CVI_TDE_COLOR_FORMAT_MP1_YCbCr420MBP,
CVI_TDE_COLOR_FORMAT_MP2_YCbCr420MBP,
CVI_TDE_COLOR_FORMAT_MP2_YCbCr420MBI,
CVI_TDE_COLOR_FORMAT_JPG_YCbCr420MBP,
CVI_TDE_COLOR_FORMAT_JPG_YCbCr444MBP,
CVI_TDE_COLOR_FORMAT_MAX
} cvi_tde_buffer_format_t;

```

【成员】

成员	描述
CVI_TDE_COLOR_FORMAT_RGB444	RGB444 格式
CVI_TDE_COLOR_FORMAT_BGR444	BGR444
CVI_TDE_COLOR_FORMAT_RGB555	RGB555 格式
CVI_TDE_COLOR_FORMAT_BGR555	BGRX8888 格式
CVI_TDE_COLOR_FORMAT_RGB565	RGB565 格式
CVI_TDE_COLOR_FORMAT_BGR565	BGR565 格式
CVI_TDE_COLOR_FORMAT_RGB888	RGB888 格式
CVI_TDE_COLOR_FORMAT_BGR888	BGR888
CVI_TDE_COLOR_FORMAT_ARGB4444	ARGB4444 格式
CVI_TDE_COLOR_FORMAT_ABGR4444	ABGR4444
CVI_TDE_COLOR_FORMAT_RGBA4444	RGBA4444
CVI_TDE_COLOR_FORMAT_BGRA4444	BGRA4444
CVI_TDE_COLOR_FORMAT_ARGB1555	ARGB1555 格式
CVI_TDE_COLOR_FORMAT_ABGR1555	ABGR1555
CVI_TDE_COLOR_FORMAT_RGBA5551	RGBA1555
CVI_TDE_COLOR_FORMAT_BGRA5551	BGRA1555
CVI_TDE_COLOR_FORMAT_ARGB8565	ARGB8565 格式
CVI_TDE_COLOR_FORMAT_ABGR8565	ABGR8565
CVI_TDE_COLOR_FORMAT_RGBA8565	RGBA8565
CVI_TDE_COLOR_FORMAT_BGRA8565	BGRA8565
CVI_TDE_COLOR_FORMAT_ARGB8888	ARGB8888 格式
CVI_TDE_COLOR_FORMAT_ABGR8888	ABGR8888
CVI_TDE_COLOR_FORMAT_RGBA8888	RGBA8888
CVI_TDE_COLOR_FORMAT_BGRA8888	BGRA8888

下页继续

表 4.2 – 续上页

成员	描述
CVI_TDE_COLOR_FORMAT_RABG8888	RABG8888
CVI_TDE_COLOR_FORMAT_CLUT1	CLUT1 格式
CVI_TDE_COLOR_FORMAT_CLUT2	CLUT2 格式
CVI_TDE_COLOR_FORMAT_CLUT4	CLUT4 格式
CVI_TDE_COLOR_FORMAT_CLUT8	CLUT8 格式
CVI_TDE_COLOR_FORMAT_ACLUT44	ACLUT44 格式
CVI_TDE_COLOR_FORMAT_ACLUT88	ACLUT88 格式
CVI_TDE_COLOR_FORMAT_A1	A1 格式
CVI_TDE_COLOR_FORMAT_A8	A8 格式
CVI_TDE_COLOR_FORMAT_YCbCr888	YCbCr888 格式
CVI_TDE_COLOR_FORMAT_AYCbCr8888	AYCbCr8888 格式
CVI_TDE_COLOR_FORMAT_YCbCr422	YCbCr422 格式, YUYV 排列
CVI_TDE_COLOR_FORMAT_PKGYYVU	YCbCr422 格式, YVYU 排列
CVI_TDE_COLOR_FORMAT_PKGUYVY	UYVY 排列
CVI_TDE_COLOR_FORMAT_PKGVYUY	VYUY 排列
CVI_TDE_COLOR_FORMAT_PKGVUYY	VUYY 排列
CVI_TDE_COLOR_FORMAT_PKGYYUV	YYUV 排列
CVI_TDE_COLOR_FORMAT_PKGVVYY	UVYY 排列
CVI_TDE_COLOR_FORMAT_PKGYYVU	YYVU 排列
CVI_TDE_COLOR_FORMAT_JPG_YCbCr400MBP	YCbCr400MBP
CVI_TDE_COLOR_FORMAT_JPG_YCbCr422MBHP	YCbCr422MBHP
CVI_TDE_COLOR_FORMAT_JPG_YCbCr422MBVP	YCbCr422MBVP
CVI_TDE_COLOR_FORMAT_MP1_YCbCr420MBP	YCbCr420MBP
CVI_TDE_COLOR_FORMAT_MP2_YCbCr420MBP	YCbCr420MBP
CVI_TDE_COLOR_FORMAT_MP2_YCbCr420MBI	YCbCr420MBI
CVI_TDE_COLOR_FORMAT_JPG_YCbCr420MBP	YCbCr420MBP
CVI_TDE_COLOR_FORMAT_JPG_YCbCr444MBP	YCbCr444MBP
CVI_TDE_COLOR_FORMAT_MAX	无效的像素格式

【注意事项】

- 目标格式不支持 CVI_TDE_COLOR_FORMAT_YCbCr400MBP, CVI_TDE_COLOR_FORMAT_YCbCr422MBHP, CVI_TDE_COLOR_FORMAT_YCbCr422MBVP, CVI_TDE_COLOR_FORMAT_YCbCr420MBP, CVI_TDE_COLOR_FORMAT_YCbCr420MBI, CVI_TDE_COLOR_FORMAT_YCbCr444MBP。
- CVI_TDE_COLOR_FORMAT_A1, CVI_TDE_COLOR_FORMAT_A8 只支持 DMA 搬移, 且不能转成其他格式。
- CLUT 格式作为输出时, 输入格式只能与输出保持一致, 且只能做拷贝操作。且不能两个源同为 CLUT 格式。
- 上述提供的 420 宏块格式排列一致, 与 `cvi_tde_mb_color_format` 的 420 宏块格式无区别。

【相关数据类型及接口】

无

4.2.2 cvi_tde_mb_color_format

【说明】

操作矩形结构体。

【定义】

```
typedef enum {
    CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr400MBP = 0,
    CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr422MBHP,
    CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr422MBVP,
    CVI_TDE_MB_COLOR_FORMAT_MP1_YCbCr420MBP,
    CVI_TDE_MB_COLOR_FORMAT_MP2_YCbCr420MBP,
    CVI_TDE_MB_COLOR_FORMAT_MP2_YCbCr420MBI,
    CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr420MBP,
    CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr444MBP,
    CVI_TDE_MB_COLOR_FORMAT_MAX
} cvi_tde_mb_color_format;
```

【成员】

成员	描述
CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr400MBP	JPEG 编码 400 宏块
CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr422MBHP	JPEG 编码 422 宏块（水平采样一半）
CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr422MBVP	JPEG 编码 422 宏块（垂直采样一半）
CVI_TDE_MB_COLOR_FORMAT_MP1_YCbCr420MBP	MPEG-1 编码 420 宏块
CVI_TDE_MB_COLOR_FORMAT_MP2_YCbCr420MBP	MPEG-2 编码 420 宏块
CVI_TDE_MB_COLOR_FORMAT_MP2_YCbCr420MBI	MPEG-2 编码 420 宏块（隔行）
CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr420MBP	JPEG 编码 420 宏块
CVI_TDE_MB_COLOR_FORMAT_JPG_YCbCr444MBP	JPEG 编码 444 宏块

【注意事项】

上述提供的 420 宏块格式排列一致，cvi_tde_color_format 的 420 宏块格式无区别。具体使用看使用的功能接口的参数类型。

【相关数据类型及接口】

无

4.2.3 cvi_tde_surface

【说明】

位图 surface 结构体。

【定义】

```
typedef struct {  
    td_phys_addr_t phys_addr;  
    CVI_U32 phys_len;  
    cvi_tde_color_format color_format;  
    CVI_U32 height;  
    CVI_U32 width;  
    CVI_U32 stride;  
    CVI_BOOL is_ycbcr_clut;  
    CVI_BOOL alpha_max_is_255;  
    CVI_BOOL support_alpha_ex_1555;  
    CVI_U8 alpha0;  
    CVI_U8 alpha1;  
    td_phys_addr_t cbcr_phys_addr;  
    CVI_U32 cbcr_phys_len;  
    CVI_U32 cbcr_stride;  
    td_phys_addr_t clut_phys_addr;  
    CVI_U32 clut_phys_len;  
} cvi_tde_surface;
```

【成员】

成员	描述
phys_addr	位图首地址。
phys_len	内存长度。
color_format	位图格式。
height	位图高度。
width	位图宽度。
strid	位图跨度。
is_ycbcr_clut	Clut 表是否位于 YCbCr 空间。
alpha_max_is_255	位图 alpha 最大值为 255 还是 128。
support_alpha_ex_1555	是否使能 1555 的 Alpha 扩展。当位图格式为 ARGB1555 时，该项有效。如果该值为 TD_FALSE，则位图像素要么全透要么全不透。
alpha0	Alpha0 值。取值范围：[0, 255]。当位图格式为 ARGB1555 且 support_alpha_ex_1555 为 TRUE 时，该项有效。在 ARGB1555 格式下，当像素的最高位为 0 时，选择该值作为 alpha 叠加的 alpha 值。
alpha1	Alpha1 值。取值范围：[0, 255]。当位图格式为 ARGB1555 且 support_alpha_ex_1555 为 TRUE 时，该项有效。在 ARGB1555 格式下，当像素的最高位为 1 时，选择该值作为 alpha 叠加的 alpha 值。
cbr_phys_addr	CbCr 分量地址。
cbr_phys_len	CbCr 内存长度。
cbr_stride	CbCr 分量跨度。
clut_phys_addr	Clut 表首地址，用作颜色扩展或颜色校正。
clut_phys_len	Clut 表的内存长度

【注意事项】

- 像素格式大于等于 BYTE 的位图格式的位图首地址和 Stride 必须按照像素格式对齐，像素格式不足 BYTE 的位图格式的位图首地址和 Stride 需要按照 BYTE 对齐。
- 像素格式不足 BYTE 的位图格式的水平起始位置和宽度必须按照 BYTE 对齐。
- YCbCr422 格式的位图的水平起始位置和宽度必须为偶数。
- CLUT 到真彩色 ARGB 的扩展是依靠检索 CLUT 表来实现的。故颜色扩展功能（如 CLUT1 格式位图扩展到 ARGB8888 格式位图）或颜色校正时，需要配置 Clut 表首地址 clut_phys_addr，且其指向的 CLUT 表内存必须是物理连续的。
- stride 最大不超过 65535。

【相关数据类型及接口】

无

4.2.4 cvi_tde_mb_surface

【说明】

变换矩阵结构体。

【定义】

```
typedef struct {
    cvi_tde_mb_color_format mb_color_format;
    td_phys_addr_t y_addr;
    CVI_U32 y_len;
    CVI_U32 y_width;
    CVI_U32 y_height;
    CVI_U32 y_stride;
    td_phys_addr_t cbc_r_phys_addr;
    CVI_U32 cbc_r_phys_len;
    CVI_U32 cbc_r_stride;
} cvi_tde_mb_surface;
```

【成员】

成员	描述
mb_color_format	宏块格式
y_addr	亮度块物理首地址
y_len	亮度内存长度
y_width	亮度块的宽度
y_height	亮度块的高度
y_stride	亮度块相邻两行的跨度
cbc_r_phys_addr	色度块的物理首地址
cbc_r_phys_len	色度内存长度
cbc_r_stride	色度块相邻两行的跨度

【注意事项】

y_stride 和 cbc_r_stride 最大不超过 65535，位图 height 和位图 width 最大不超过 8192。

【相关数据类型及接口】

无

4.2.5 cvi_tde_rect

【说明】

TDE 操作区域属性。

【定义】

```
typedef struct {
    CVI_S32 pos_x;
    CVI_S32 pos_y;
```

(下页继续)

(续上页)

```
CVI_U32 width;
CVI_U32 height;
} cvi_tde_rect;
```

【成员】

成员	描述
pos_x	操作区域的起始横坐标，以像素数为单位；有效范围：[0, 位图宽度)
pos_y	操作区域的起始纵坐标，以像素数为单位；有效范围：[0, 位图高度)
width	操作区域的宽度，以像素数为单位。有效范围：(0, 4096]
height	操作区域的高度，以像素数为单位。有效范围：(0, 4096]

【注意事项】

- 操作区域与位图的关系如:ref: 图 2-2 < 图 2-2> 所示。
- 若操作区域与位图部分相交，则取相交部分为实际参与操作的区域；若操作区域与位图不相交，则返回相应错误码。
- rect.pos_x 加上 rect.width 大于 surface.width 会重新计算 rect.width= surface.width-rect.pos_x。
- rect.pos_y 加上 rect.height 大于 surface.height 会重新计算 rect.height =surface.height - rect.pos_y。

【相关数据类型及接口】

无

4.2.6 cvi_tde_none_src

【说明】

TDE 操作组合类型（按源的个数区分）。此为 0 源操作。

【定义】

```
typedef struct {
    cvi_tde_surface *dst_surface;
    cvi_tde_rect *dst_rect;
} cvi_tde_none_src;
```

【成员】

成员	描述
dst_surface	目标 surface 结构体
dst_rect	目标区域

【注意事项】

无

【相关数据类型及接口】

无

4.2.7 cvi_tde_single_src

【说明】

TDE 操作组合类型（按源的个数区分）。此为单源操作。

【定义】

```
typedef struct {  
    cvi_tde_surface *src_surface;  
    cvi_tde_surface *dst_surface;  
    cvi_tde_rect *src_rect;  
    cvi_tde_rect *dst_rect;  
} cvi_tde_single_src;
```

【成员】

成员	描述
src_surface	源 surface 结构体
dst_surface	目标 surface 结构体
src_rect	源区域
dst_rect	目标区域

【注意事项】

无

【相关数据类型及接口】

无

4.2.8 cvi_tde_mb_src

【说明】

TDE 操作组合类型（按源的个数区分）。此为双源操作。

【定义】

```
typedef struct {  
    cvi_tde_mb_surface *mb_surface;  
    cvi_tde_surface *dst_surface;  
    cvi_tde_rect *src_rect;  
    cvi_tde_rect *dst_rect;  
} cvi_tde_mb_src;
```

【成员】

成员	描述
mb_surface	源 surface 结构体
dst_surface	目标 surface 结构体
src_rect	源区域
dst_rect	目标区域

【注意事项】

无

【相关数据类型及接口】

无

4.2.9 cvi_tde_double_src

【说明】

TDE 操作组合类型（按源的个数区分）。此为双源操作。

【定义】

```
typedef struct {  
    cvi_tde_surface *bg_surface;  
    cvi_tde_surface *fg_surface;  
    cvi_tde_surface *dst_surface;  
    cvi_tde_rect *bg_rect;  
    cvi_tde_rect *fg_rect;  
    cvi_tde_rect *dst_rect;  
} cvi_tde_double_src;
```

【成员】

成员	描述
bg_surface	背景 surface 结构体。
fg_surface	前景 surface 结构体。
dst_surface	目标 surface 结构体。
bg_rect	背景区域
fg_rect	前景区域
dst_rect	目标区域

【注意事项】

无

【相关数据类型及接口】

无

4.2.10 cvi_tde_alpha_blending

【说明】

逻辑运算类型属性。

【定义】

```
typedef enum {  
    CVI_TDE_ALPHA_BLENDING_NONE = 0x0,  
    CVI_TDE_ALPHA_BLENDING_BLEND = 0x1,  
    CVI_TDE_ALPHA_BLENDING_ROP = 0x2,  
    CVI_TDE_ALPHA_BLENDING_COLORIZE = 0x4,  
    CVI_TDE_ALPHA_BLENDING_MAX = 0x8  
} cvi_tde_alpha_blending;
```

【成员】

成员	描述
CVI_TDE_ALPHA_BLENDING_NONE	无逻辑运算
CVI_TDE_ALPHA_BLENDING_BLEND	Alpha 叠加类型
CVI_TDE_ALPHA_BLENDING_ROP	布尔运算类型
CVI_TDE_ALPHA_BLENDING_COLORIZE	Colorize 操作
CVI_TDE_ALPHA_BLENDING_MAX	无效逻辑运算类型

【注意事项】

若进行两个位图的 alpha 叠加运算，请选择 CVI_TDE_ALPHA_BLENDING_BLEND，若进行 Colorize 操作，请选择 CVI_TDE_ALPHA_BLENDING_COLORIZE。如选择 CVI_TDE_ALPHA_BLENDING_ROP，指进行布尔逻辑运算类型。通过指定 cvi_tde_opt 结构中的 rop_color 和 rop_alpha 成员，分别指定颜色和 alpha 分量的 ROP 运算类型。

【相关数据类型及接口】

无

4.2.11 cvi_tde_rop_mode

【说明】

TDE 支持的 ROP 操作类型。

【定义】

```
typedef enum {  
    CVI_TDE_ROP_BLACK = 0,  
    CVI_TDE_ROP_NOTMERGEPEN,  
    CVI_TDE_ROP_MASKNOTPEN,  
    CVI_TDE_ROP_NOTCOPYPEN,
```

(下页继续)

(续上页)

```

CVI_TDE_ROP_MASKPENNOT,
CVI_TDE_ROP_NOT,
CVI_TDE_ROP_XORPEN,
CVI_TDE_ROP_NOTMASKPEN,
CVI_TDE_ROP_MASKPEN,
CVI_TDE_ROP_NOTXORPEN,
CVI_TDE_ROP_NOP,
CVI_TDE_ROP_MERGEENOTPEN,
CVI_TDE_ROP_COPYPEN,
CVI_TDE_ROP_MERGEENOT,
CVI_TDE_ROP_MERGEEN,
CVI_TDE_ROP_WHITE,
CVI_TDE_ROP_MAX
} cvi_tde_rop_mode;

```

【成员】

成员	描述
CVI_TDE_ROP_BLACK	Blackness
CVI_TDE_ROP_NOTMERGEEN	$\sim(S2+S1)$
CVI_TDE_ROP_MASKNOTPEN	$\sim S2 \& S1$
CVI_TDE_ROP_NOTCOPYPEN	$\sim S2$
CVI_TDE_ROP_MASKPENNOT	$S2 \& \sim S1$
CVI_TDE_ROP_NOT	$\sim S1$
CVI_TDE_ROP_XORPEN	$S2 \wedge S1$
CVI_TDE_ROP_NOTMASKPEN	$\sim(S2 \& S1)$
CVI_TDE_ROP_MASKPEN	$S2 \& S1$
CVI_TDE_ROP_NOTXORPEN	$\sim(S2 \wedge S1)$
CVI_TDE_ROP_NOP	$S1$
CVI_TDE_ROP_MERGEENOTPEN	$\sim S2 + S1$
CVI_TDE_ROP_COPYPEN	$S2$
CVI_TDE_ROP_MERGEENOT	$S2 + \sim S1$
CVI_TDE_ROP_MERGEEN	$S2 + S1$
CVI_TDE_ROP_WHITE	Whiteness
CVI_TDE_ROP_MAX	无效的 ROP 类型

【注意事项】

S1 表示背景位图，S2 表示位前景位图。

不同操作时，S1、S2 具体所指的位图不同，详见每个接口的说明部分。

如果对两幅位图进行的操作类型选择为 CVI_TDE_ALPHA_BLENDING_ROP，则可以分别对颜色空间和 Alpha 指定不同的 ROP 操作。

例如：设前景位图和背景位图均为 ARGB8888 格式的位图。

前景位图的像素值为 foreground，背景位图的像素值为 background，经过运算后的像素的值为 pixel，取 Alpha 的 ROP 操作为 Whiteness，取颜色空间的 ROP 操作为 Blackness，则经过运算之后的像素值 pixel 计算如下：

```
pixel.alpha = 0xff;
```

```
pixel.r = pixel.g = pixel.b = 0x00;
```

其中 pixel.alpha, pixel.r, pixel.g, pixel.b 分别代表运算后位图的各个分量。

【相关数据类型及接口】

无

4.2.12 cvi_tde_mirror_mode

【说明】

图像镜像属性。

【定义】

```
typedef enum {  
    CVI_TDE_MIRROR_NONE = 0,  
    CVI_TDE_MIRROR_HOR,  
    CVI_TDE_MIRROR_VER,  
    CVI_TDE_MIRROR_BOTH,  
    CVI_TDE_MIRROR_MAX  
} cvi_tde_mirror_mode;
```

【成员】

成员	描述
CVI_TDE_MIRROR_NONE	输出图像不进行镜像操作
CVI_TDE_MIRROR_HOR	输出图像水平镜像
CVI_TDE_MIRROR_VER	输出图像垂直镜像
CVI_TDE_MIRROR_BOTH	输出图像水平 + 垂直镜像
CVI_TDE_MIRROR_MAX	S 无效的镜像类型

【注意事项】

无

【相关数据类型及接口】

无

4.2.13 cvi_tde_clip_mode

【说明】

剪切操作类型。

【定义】

```
typedef enum {  
    CVI_TDE_CLIP_MODE_NONE = 0,  
    CVI_TDE_CLIP_MODE_INSIDE,  
    CVI_TDE_CLIP_MODE_OUTSIDE,  
}
```

(下页继续)

(续上页)

```
CVI_TDE_CLIP_MODE_MAX
} cvi_tde_clip_mode;
```

【成员】

成员	描述
CVI_TDE_CLIP_MODE_NONE	输出结果不做剪切
CVI_TDE_CLIP_MODE_INSIDE	区域内剪切模式
CVI_TDE_CLIP_MODE_OUTSIDE	区域外剪切模式
CVI_TDE_CLIP_MODE_MAX	无效的剪切模式

【注意事项】

无

【相关数据类型及接口】

无。

4.2.14 cvi_tde_mb_resize

【说明】

宏块格式缩放类型。

【定义】

```
typedef enum {
    CVI_TDE_MB_RESIZE_NONE = 0,
    CVI_TDE_MB_RESIZE_QUALITY_LOW,
    CVI_TDE_MB_RESIZE_QUALITY_MIDDLE,
    CVI_TDE_MB_RESIZE_QUALITY_HIGH,
    CVI_TDE_MB_RESIZE_MAX
} cvi_tde_mb_resize;
```

【成员】

成员	描述
CVI_TDE_MB_RESIZE_NONE	不进行缩放
CVI_TDE_MB_RESIZE_QUALITY_LOW	宏块 Suface 的低质量缩放模式
CVI_TDE_MB_RESIZE_QUALITY_MIDDLE	宏块 Suface 的中质量缩放模式
CVI_TDE_MB_RESIZE_QUALITY_HIGH	宏块 Suface 的高质量缩放模式
CVI_TDE_MB_RESIZE_MAX	无效的缩放模式

【注意事项】

无

【相关数据类型及接口】

无

4.2.15 cvi_tde_fill_color

【说明】

图像填充色属性结构。

【定义】

```
typedef struct {  
    cvi_tde_color_format color_format;  
    CVI_U32 color_value;  
} cvi_tde_fill_color;
```

【成员】

成员	描述
color_format	填充色格式
color_value	填充值

【注意事项】

填充值必须与填充格式相匹配，如某位图需填充为蓝色，可指定填充色格式为 ARGB1555，填充值为 0x801F（此处 alpha 位为 1）。

【相关数据类型及接口】

无。

4.2.16 cvi_tde_colorkey_mode

【说明】

TDE colorkey 模式属性。

【定义】

```
typedef enum {  
    CVI_TDE_COLORKEY_MODE_NONE = 0,  
    CVI_TDE_COLORKEY_MODE_FG,  
    CVI_TDE_COLORKEY_MODE_BG,  
    CVI_TDE_COLORKEY_MODE_MAX  
} cvi_tde_colorkey_mode;
```

【成员】

成员	描述
CVI_TDE_COLORKEY_MODE_NONE	不做 colorkey 操作
CVI_TDE_COLORKEY_MODE_FG	对前景位图进行 colorkey 操作
CVI_TDE_COLORKEY_MODE_BG	对背景位图进行 colorkey 操作
CVI_TDE_COLORKEY_MODE_MAX	无效的 colorkey 模式

【注意事项】

对前景位图进行 colorkey 时，对于颜色扩展，在 CLUT 前做 colorkey；对于颜色校正，在 CLUT 后做 colorkey。

【相关数据类型及接口】

无

4.2.17 cvi_tde_colorkey_component

【说明】

单个颜色分量的关键色属性。

【定义】

```
typedef struct {  
    CVI_U8 min_component;  
    CVI_U8 max_component;  
    CVI_U8 is_component_out;  
    CVI_U8 is_component_ignore;  
    CVI_U8 component_mask;  
} cvi_tde_colorkey_component;
```

【成员】

成员	描述
min_component	分量关键色最小值
max_component	分量关键色最大值
is_component_out	分量关键色在范围内或范围外
is_component_ignore	分量是否忽略
component_mask	分量掩码

【注意事项】

- 结构成员 is_component_ignore 指关键色比较时是否忽略该分量的比较，而总是认为其满足关键色要求。
 - 若 is_component_ignore 为 TRUE，表示关键色比较时，忽略该分量的比较，认为该分量总是满足关键色要求。
 - 若 is_component_ignore 为 FALSE，表示需要根据 [最小关键色，最大关键色] 范围以及属性 is_component_out 判断该分量的值是否符合关键色要求。
- 结构体成员 component_mask 控制分量中的哪些 bit 位有效，即分量会和 component_mask 做与运算。若 component_mask 为 0 则分量的值为 0，若 component_mask 为 0xFF 则分量的值为该像素点的分量值，以此类推。

【相关数据类型及接口】

无

4.2.18 cvi_tde_colorkey

【说明】

关键色属性

【定义】

```
typedef union {  
    struct {  
        cvi_tde_colorkey_component alpha;  
        cvi_tde_colorkey_component red;  
        cvi_tde_colorkey_component green;  
        cvi_tde_colorkey_component blue;  
    } argb_colorkey;  
    struct {  
        cvi_tde_colorkey_component alpha;  
        cvi_tde_colorkey_component y;  
        cvi_tde_colorkey_component cb;  
        cvi_tde_colorkey_component cr;  
    } ycbcr_colorkey;  
    struct {  
        cvi_tde_colorkey_component alpha;  
        cvi_tde_colorkey_component clut;  
    } clut_colorkey;  
} cvi_tde_colorkey;
```

【成员】

结构 argb_colorkey 成员：表示位图格式为 ARGB 类型时各分量的关键色属性。

成员	描述
alpha	alpha 分量关键色属性
red	Red 分量关键色属性
green	Green 分量关键色属性
is_cobluecomponent_ignore	Blue 分量关键色属性

结构 ycbcr_colorkey 成员：表示位图格式为 AYCbCr 类型时，各分量的关键色属性。

成员	描述
alpha	alpha 分量关键色属性
y	Y 分量关键色属性
cb	Cb 分量关键色属性
cr	Cr 分量关键色属性

结构 clut_colorkey 成员：表示位图格式为 CLUT 类型时，各分量的关键色属性。

成员	描述
alpha	alpha 分量关键色属性
clut	CLUT 分量关键色属性

联合类型 CVI_TDE_COLORkey 成员，各分量的关键色属性。

成员	描述
argb_colorkey	当位图格式为 ARGB 类型时，关键色属性
ycbcr_colorkey	位图格式为 AYCbCr 类型时，关键色属性
clut_colorkey	当位图格式为 CLUT 类型时，关键色属性

【注意事项】

不管当前位图是什么格式，color space 颜色上限值和下限值都应该是 ARGB8888 格式的值。

【相关数据类型及接口】

无

4.2.19 cvi_tde_out_alpha_from

【说明】

输出 alpha 来源类型。

【定义】

```
typedef enum {
    CVI_TDE_OUT_ALPHA_FROM_NORM = 0,
    CVI_TDE_OUT_ALPHA_FROM_BG,
    CVI_TDE_OUT_ALPHA_FROM_FG,
    CVI_TDE_OUT_ALPHA_FROM_GLOBALALPHA,
    CVI_TDE_OUT_ALPHA_FROM_MAX
} cvi_tde_out_alpha_from;
```

【成员】

成员	描述
CVI_TDE_OUT_ALPHA_FROM_NORM	输出图像的 alpha 来源于 alpha blending 的结果或者抗闪烁的结果
CVI_TDE_OUT_ALPHA_FROM_BG	输出图像的 alpha 来源于背景位图
CVI_TDE_OUT_ALPHA_FROM_FG	输出图像的 alpha 来源于前景位图
CVI_TDE_OUT_ALPHA_FROM_GLOBALALPHA	输出图像的 alpha 来源于全局 alpha
CVI_TDE_OUT_ALPHA_FROM_MAX	无效值

【注意事项】

无

【相关数据类型及接口】

无

4.2.20 cvi_tde_filter_mode

【说明】

图像滤波模式。

【定义】

```
typedef enum {  
    CVI_TDE_FILTER_MODE_COLOR = 0,  
    CVI_TDE_FILTER_MODE_ALPHA,  
    CVI_TDE_FILTER_MODE_BOTH,  
    CVI_TDE_FILTER_MODE_NONE,  
    CVI_TDE_FILTER_MODE_MAX  
} cvi_tde_filter_mode;
```

【成员】

成员	描述
CVI_TDE_FILTER_MODE_COLOR	对颜色进行滤波
CVI_TDE_FILTER_MODE_ALPHA	对 alpha 通道滤波
CVI_TDE_FILTER_MODE_BOTH	对颜色和 alpha 通道同时滤波
CVI_TDE_FILTER_MODE_NONE	不进行滤波
CVI_TDE_FILTER_MODE_MAX	无效的滤波模式

【注意事项】

图像的缩放或抗闪都是一种滤波，故在图像缩放或（和）抗闪操作时，需要指明滤波模式。

【相关数据类型及接口】

无

4.2.21 cvi_tde_blend_mode

【说明】

用户自定义 alpha 混合模式。

【定义】

```
typedef enum {  
    CVI_TDE_BLEND_ZERO = 0x0,  
    CVI_TDE_BLEND_ONE,  
    CVI_TDE_BLEND_SRC2COLOR,  
    CVI_TDE_BLEND_INVSR2COLOR,  
    CVI_TDE_BLEND_SRC2ALPHA,  
    CVI_TDE_BLEND_INVSR2ALPHA,  
    CVI_TDE_BLEND_SRC1COLOR,  
    CVI_TDE_BLEND_INVSR1COLOR,  
    CVI_TDE_BLEND_SRC1ALPHA,  
    CVI_TDE_BLEND_INVSR1ALPHA,  
    CVI_TDE_BLEND_SRC2ALPHASAT,  
    CVI_TDE_BLEND_INVSR2ALPHASAT,  
    CVI_TDE_BLEND_SRC1ALPHASAT,  
    CVI_TDE_BLEND_INVSR1ALPHASAT,  
    CVI_TDE_BLEND_MAX  
}
```

(下页继续)

(续上页)

```
CVI_TDE_BLEND_MAX
} cvi_tde_blend_mode;
```

【成员】

$\text{pixel} = (\text{foreground} \times \text{fs} + \text{background} \times \text{fd})$ ，其中：

- fs: foreground blend coefficient。
- fd: destination blend coefficient。
- pixel: 运算以后的像素值。
- foreground: 前景位图的像素值。
- background: 背景位图的像素值。
- sa: foreground alpha。
- da: background alpha。
- sc: foreground color。
- dc: background color。
- sc 为颜色值归一化后的值，各分量分开计算。即颜色分量归一化值计算公式为: 各分量颜色值/255
- fs 和 fd 分别为源位图像素和目的位图像素的系数，选择下表中不同的项代表不同的系数。

成员	描述
CVI_TDE_BLEND_ZERO	0
CVI_TDE_BLEND_ONE	1
CVI_TDE_BLEND_SRC2COLOR	取 sc
CVI_TDE_BLEND_INVSR2COLOR	取 1-sc
CVI_TDE_BLEND_SRC2ALPHA	sa
CVI_TDE_BLEND_INVSR2ALPHA	1-sa
CVI_TDE_BLEND_SRC1COLOR	dc
CVI_TDE_BLEND_INVSR1COLOR	1-dc
CVI_TDE_BLEND_SRC1ALPHA	da
CVI_TDE_BLEND_INVSR1ALPHA	1-da
CVI_TDE_BLEND_SRC2ALPHASAT	$\min(1-\text{da}, \text{sa}) + 1$
CVI_TDE_BLEND_MAX	无效的 alpha 混合模式

【注意事项】

在前景位图和背景位图作叠加运算时，可以分别设置 Src1 通道和 Src2 通道的叠加模式。现在支持 11 种叠加模式。当 `cvi_tde_blend_mode` 取 `cvi_tde_blend_cmd` 时，可通过取 `cvi_tde_blend_mode` 里面不同的项实现不同类型的叠加。

【相关数据类型及接口】

无

4.2.22 cvi_tde_blend_cmd

【说明】

Alpha 混合命令。用于计算进行 alpha 混合以后的像素值。

【定义】

```
typedef enum {  
    CVI_TDE_BLEND_CMD_NONE = 0x0,  
    CVI_TDE_BLEND_CMD_CLEAR,  
    CVI_TDE_BLEND_CMD_SRC,  
    CVI_TDE_BLEND_CMD_SRCOVER,  
    CVI_TDE_BLEND_CMD_DSTOVER,  
    CVI_TDE_BLEND_CMD_SRCIN,  
    CVI_TDE_BLEND_CMD_DSTIN,  
    CVI_TDE_BLEND_CMD_SRCOUT,  
    CVI_TDE_BLEND_CMD_DSTOUT,  
    CVI_TDE_BLEND_CMD_SRCATOP,  
    CVI_TDE_BLEND_CMD_DSTATOP,  
    CVI_TDE_BLEND_CMD_ADD,  
    CVI_TDE_BLEND_CMD_XOR,  
    CVI_TDE_BLEND_CMD_DST,  
    CVI_TDE_BLEND_CMD_CONFIG,  
    CVI_TDE_BLEND_CMD_MAX  
} cvi_tde_blend_cmd;
```

【成员】

$\text{pixel} = (\text{foreground} \times \text{fs} + \text{background} \times \text{fd})$, 其中:

- fs: foreground blend coefficient。
- fd: destination blend coefficient。
- pixel: 运算以后的像素值。
- foreground: 前景位图的像素值。
- background: 背景位图的像素值。
- sa: foreground alpha。
- da: background alpha。

成员	描述
CVI_TDE_BLEND_CMD_NONE	fs 取 sa, fd 取 1.0-sa
CVI_TDE_BLEND_CMD_CLEAR	fs 取 0.0, fd 取 0.0
CVI_TDE_BLEND_CMD_SRC	fs 取 1.0, fd 取 0.0
CVI_TDE_BLEND_CMD_SRCOVER	fs 取 1.0, fd 取 1.0-sa
CVI_TDE_BLEND_CMD_DSTOVER	fs 取 1.0-da, fd 取 1.0
CVI_TDE_BLEND_CMD_SRCIN	fs 取 da, fd 取 0.0
CVI_TDE_BLEND_CMD_DSTIN	fs 取 0.0, fd 取 sa
CVI_TDE_BLEND_CMD_SRCOUT	fs 取 1.0-da, fd 取 0.0
CVI_TDE_BLEND_CMD_DSTOUT	fs 取 0.0, fd 取 1.0-sa
CVI_TDE_BLEND_CMD_SRCATOP	fs 取 da, fd 取 1.0-sa
CVI_TDE_BLEND_CMD_DSTATOP	fs 取 1.0-da, fd 取 sa
CVI_TDE_BLEND_CMD_ADD	dfs 取 1.0, fd 取 1.0
CVI_TDE_BLEND_CMD_XOR	fs 取 1.0-da, fd 取 1.0-sa
CVI_TDE_BLEND_CMD_DST	fs 取 0.0, fd 取 1.0
CVI_TDE_BLEND_CMD_CONFIG	用户自己配置参数
CVI_TDE_BLEND_CMD_MAX	无效的 alpha 混合命令

【举例】**【注意事项】**

clut 相关格式不支持 alpha 混合。

【相关数据类型及接口】

无

4.2.23 cvi_tde_blend_opt

【说明】

alpha 混合操作选项。

【定义】

```
typedef struct {
    CVI_BOOL global_alpha_en;
    CVI_BOOL pixel_alpha_en;
    CVI_BOOL src1_alpha_premulti;
    CVI_BOOL src2_alpha_premulti;
    cvi_tde_blend_cmd blend_cmd;
    cvi_tde_blend_mode src1_blend_mode;
    cvi_tde_blend_mode src2_blend_mode;
} cvi_tde_blend_opt;
```

【成员】

成员	描述
global_alpha_en	是否使能全局 alpha
pixel_alpha_en	是否使能像素 alpha
src1_alpha_premulti	是否使能 Src1 alpha 预乘
src2_alpha_premulti	是否使能 Src2 alpha 预乘
blend_cmd	alpha 混合命令
src1_blend_mode	src1 blend 模式选择, 在 blend_cmd = CVI_TDE_BLEND_CMD_CONFIG 时有效
src2_blend_mode	src2 blend 模式选择, 在 blend_cmd = CVI_TDE_BLEND_CMD_CONFIG 时有效

【注意事项】

- clut 相关格式不支持 alpha 混合;
- 源 1 (Src1) 是背景。

【相关数据类型及接口】

无

4.2.24 cvi_tde_csc_opt

【说明】

CSC 参数选项。

【定义】

```
typedef struct {  
    CVI_BOOL src_csc_user_en;  
    CVI_BOOL src_csc_param_reload_en;  
    CVI_BOOL dst_csc_user_en;  
    CVI_BOOL dst_csc_param_reload_en;  
    td_phys_addr_t src_csc_param_addr;  
    CVI_S32 src_csc_param_len;  
    td_phys_addr_t dst_csc_param_addr;  
    CVI_S32 dst_csc_param_len;  
} cvi_tde_csc_opt;
```

【成员】

成员	描述
src_csc_user_en	用户自定义 ICSC 参数使能
src_csc_param_reload_en	重新加载用户自定义 ICSC 参数使能
dst_csc_user_en	用户自定义 OCSC 参数使能
dst_csc_param_reload_en	重新加载用户自定义 OCSC 参数使能
src_csc_param_addr	ICSC 参数地址，要求 128bit 对齐
src_csc_param_len	ICSC 参数内存长度、
dst_csc_param_addr	OCSC 参数地址，要求 128bit 对齐
dst_csc_param_len	OCSC 参数内存长度

【注意事项】

无

【相关数据类型及接口】

无

4.2.25 cvi_tde_opt

【说明】

TDE 操作属性结构体。

【定义】

```
typedef struct {  
    cvi_tde_alpha_blending alpha_blending_cmd;  
    cvi_tde_rop_mode rop_color;  
    cvi_tde_rop_mode rop_alpha;  
    cvi_tde_colorkey_mode colorkey_mode;  
    cvi_tde_colorkey colorkey_value;  
    cvi_tde_clip_mode clip_mode;  
    cvi_tde_rect clip_rect;  
    CVI_BOOL resize;  
    cvi_tde_filter_mode filter_mode;  
    cvi_tde_mirror_mode mirror;  
    CVI_BOOL clut_reload;  
    CVI_U8 global_alpha;  
    cvi_tde_out_alpha_from out_alpha_from;  
    CVI_U32 color_resize;  
    cvi_tde_blend_opt blend_opt;  
    cvi_tde_csc_opt csc_opt;  
    CVI_BOOL is_compress;  
    CVI_BOOL is_decompress;  
} cvi_tde_opt;
```

【成员】

成员	描述
alpha_blending_cmd	逻辑运算类型
rop_color	颜色空间 ROP 类型
rop_alpha	Alpha 的 ROP 类型
colorkey_mode	colorkey 方式
colorkey_value	colorkey 设置值
clip_mode	区域内作 clip 还是区域外作 clip
clip_rect	clip 区域定义
resize	是否缩放
filter_mode	缩放时使用的滤波模式
mirror	镜像类型
clut_reload	是否重新加载 Clut 表
global_alpha	全局 alpha 值取值范围: [0, 255]
out_alpha_from	输出 alpha 来源
color_resize	colorize 的值
blend_opt	Alpha 混合操作选项
csc_opt	CSC 参数选项
is_compress	压缩 (tde 的压缩输出仅用于与 gfbg 模块解压配套使用, 外部设置无意义)
is_decompress	解压 (设置无效)

【功能说明】

- 按位布尔运算, 即 ROPROP 操作是指在将前景位图的 RGB 颜色分量和 alpha 分量值与背景位图的 RGB 颜色分量值和 alpha 分量值进行按位的布尔运算 (包括按位与, 按位或等), 将结果输出, 如:ref: 图 3-2 < 图 3-2> 所示。

【注意事项】

所有对光栅格式的运算, 内部均转换为 32 位数据进行运算, 再按照指定格式输出。

【相关数据类型及接口】

无

4.2.26 cvi_tde_mb_opt

【说明】

宏块 Surface 的操作属性。

【定义】

```
typedef struct {
    cvi_tde_clip_mode clip_mode;
    cvi_tde_rect clip_rect;
    cvi_tde_mb_resize resize_en;
    CVI_BOOL is_set_out_alpha;
    CVI_U8 out_alpha;
} cvi_tde_mb_opt;
```

【成员】

成员	描述
clip_mode	Clip 模式选择：区域内作 clip 还是区域外作 clip
clip_rect	Clip 区域定义
resize_en	宏块缩放模式：不缩放/高质量缩放/中质量缩放/低质量缩放
is_set_out_alpha	是否用户指定输出结果位图的 alpha 值如果不设置 Alpha，则默认输出最大 Alpha 值
out_alpha	用户指定的输出结果位图的 alpha 值

【注意事项】

无

【相关数据类型及接口】

无

4.2.27 cvi_tde_pattern_fill_opt

【说明】

模式填充操作信息定义。

【定义】

```
typedef struct {  
    cvi_tde_alpha_blending alpha_blending_cmd;  
    cvi_tde_rop_mode rop_color;  
    cvi_tde_rop_mode rop_alpha;  
    cvi_tde_colorkey_mode colorkey_mode;  
    cvi_tde_colorkey colorkey_value;  
    cvi_tde_clip_mode clip_mode;  
    cvi_tde_rect clip_rect;  
    CVI_BOOL clut_reload;  
    CVI_U8 global_alpha;  
    cvi_tde_out_alpha_from out_alpha_from;  
    CVI_U32 color_resize;  
    cvi_tde_blend_opt blend_opt;  
    cvi_tde_csc_opt csc_opt;  
} cvi_tde_pattern_fill_opt;
```

【成员】

成员	描述
alpha_blending_cmd	逻辑运算类型
rop_color	颜色空间 ROP 类型
rop_alpha	Alpha 的 ROP 类型
colorkey_mode	colorkey 方式
colorkey_value	colorkey 设置值
clip_mode	clip 模式
clip_rect	clip 区域
clut_reload	是否重载 clut 表
global_alpha	全局 alpha
out_alpha_from	输出 alpha 来源
color_resize	Colorize 值
blend_opt	Blend 操作选项
csc_opt	Csc 参数选项

【注意事项】

无

【相关数据类型及接口】

无

4.2.28 cvi_tde_rotate_angle

【说明】

旋转角度。

【定义】

```
typedef enum {  
    CVI_TDE_ROTATE_CLOCKWISE_90 = 0,  
    CVI_TDE_ROTATE_CLOCKWISE_180,  
    CVI_TDE_ROTATE_CLOCKWISE_270,  
    CVI_TDE_ROTATE_MAX  
} cvi_tde_rotate_angle;
```

【成员】

成员	描述
CVI_TDE_ROTATE_CLOCKWISE_90	顺时针旋转 90 度
CVI_TDE_ROTATE_CLOCKWISE_180	顺时针旋转 180 度
CVI_TDE_ROTATE_CLOCKWISE_270	顺时针旋转 270 度
CVI_TDE_ROTATE_MAX	无效参数

【注意事项】

无

【相关数据类型及接口】

无

4.2.29 cvi_tde_corner_rect_info

【说明】

路径中每个点信息结构体。

【定义】

```
typedef struct {  
    CVI_U32 width;  
    CVI_U32 height;  
    CVI_U32 inner_color;  
    CVI_U32 outer_color;  
} cvi_tde_corner_rect_info;
```

【成员】

成员	描述
width	角框宽
height	角框高
inner_color	角框内颜色
outer_color	角框外颜色

【注意事项】

角框内颜色和角框外颜色根据图像的格式填入一个像素值。

【相关数据类型及接口】

无

4.2.30 cvi_tde_corner_rect

【说明】

角框属性与角框绘制位置的定义。

【定义】

```
typedef struct {  
    cvi_tde_rect *corner_rect_region;  
    cvi_tde_corner_rect_info *corner_rect_info;  
} cvi_tde_corner_rect;
```

【成员】

成员	描述
corner_rect_region	角框绘制区域指针: corner_rect_region-> pos_x;x 起始坐标 corner_rect_region-> pos_y;y 起始坐标 corner_rect_region-> width; 绘制宽 corner_rect_region-> height; 绘制高
corner_rect_info	角框属性, 详见数据结构:ref:cvi_tde_corner_rect_info <cvi_tde_corner_rect_info> 介绍

【注意事项】

无

【相关数据类型及接口】

无

4.2.31 cvi_tde_line

【说明】

Drd 画线功能线条属性的定义。

【定义】

```
typedef struct {  
    CVI_S32 start_x;  
    CVI_S32 start_y;  
    CVI_S32 end_x;  
    CVI_S32 end_y;  
    CVI_U32 thick;  
    CVI_U32 color;  
} cvi_tde_line;
```

【成员】

成员	描述
start_x	起点横坐标 [-16383, 16383]
start_y	起点纵坐标 [-16383, 16383]
end_x	终点横坐标 [-16383, 16383]
end_y	终点纵坐标 [-16383, 16383]
thick	线条宽度
color	颜色值

【注意事项】

颜色值: 根据图像的格式填入一个像素值。

【相关数据类型及接口】

cvi_tde_draw_line

5 错误码

列举此软件模块接口返回的错误码

表 5.1: API 错误码表

错误代码	宏定义	描述
0	CVI_TDE_SUCCESS	成功
1	CVI_TDE_INVALID_ARGUMENT	指定了无效的参数
2	CVI_TDE_SUCCESS	成功
3	CVI_TDE_OUT_OF_MEMORY	内存溢出
4	CVI_TDE_NO_CONTEXT	未指定上下文或未实例化的上下文
5	CVI_TDE_OUT_OF_RESOURCES	系统资源不足
6	CVI_TDE_GENERIC_IO	无法与内核驱动程序通信
7	CVI_TDE_NOT_SUPPORT	不支持的函数调用
8	CVI_TDE_ALREADY_EXISTS	对象已经存在
9	CVI_TDE_NOT_ALIGNED	数据对齐错误
10	CVI_TDE_FLEXA_TIME_OUT	TDE 请求段缓冲区错误
11	CVI_TDE_FLEXA_HANDSHAKE_FAIL	TDE 和 SBI 同步器握手失败

5.1 测试

当前测试仅用于算能端测 CV186AH 平台 TDE 模块测试。

5.1.1 环境准备

硬件环境：描述测试平台位置，室内室外或者暗室，室外测试时对光线敏感的测试需明确测试所处时间段描述测试的硬件平台。

软件环境：包括但不限于操作系统、开源软件工具包、操作系统驱动程序、网络分析器，模拟器、测试覆盖和测试管理工具、或者操作系统提供的工具诸如 df、du 等。

5.1.2 单元测试

单元测试是检查代码粒度的 bug(一般是以函数和对象的方法为粒度)，测试最小粒度是否满足需求，每个模块均需要根据使用不同的参数调用这个函数，并断言它是否是期待的结果。

参考《错误码》部分的“API 错误码表”语法

5.1.3 功能测试

详细描述此软件模块功能测试项并填写到表单中。

参考《错误码》部分的“API 错误码表”语法

5.1.4 性能测试

详细描述此软件模块的性能测试项目，包括但不限于不同模式下的最大性能设置以及衡量性能的指标包括速率，码流率，帧率，以及性能制约 (限制) 因素。

参考《错误码》部分的“API 错误码表”语法