

# 基于 ZYNQ 的高清视频与图形叠加显示技术

连成哲, 翟东奇, 蒋沅嵩, 孙建华, 李家齐

(华中光电技术研究所-武汉光电国家研究中心, 湖北 武汉 430223)

**摘要:**传统的显示与控制系统在有高清视频显示需求时,往往依赖搭配专用图形处理硬件如 GPU、显卡等的工控机实现,其具有大体积、高功耗等缺点。针对这些问题,基于 ARM+FPGA 异构芯片 ZYNQ SoC 研究了高清视频与图形界面的叠加显示技术。在 ARM 端定制了嵌入式 Linux 操作系统,在系统中集成了图像采集、DRM 显示驱动以及 dma-buf 内存共享驱动,结合 Qt 图形库生成人机交互与控制界面,应用层通过 libdrm 接口控制多个显示图层的透明度与叠加关系。ARM 通过 AXI 总线与 FPGA 互联,控制逻辑资源完成图像的传输、解码、图像与 Qt 图形界面的叠加和显示等大数据量的工作,从而降低 CPU 的负载,充分发挥 ZYNQ 异构芯片优势。该高清图像与图形叠加的软硬件协同解决方案,相对于传统的显控方案,具有灵活、低功耗、体积小、低成本等优势。

**关键词:**显示技术; DRM 框架; 多图层; 叠加; 高清图像

中图分类号: TP368.2

文献标识码: A

文章编号: 1673-629X(2022)04-0074-06

doi: 10.3969/j.issn.1673-629X.2022.04.013

## Research on HD Video and Graphics Overlay Display Technology Based on ZYNQ

LIAN Cheng-zhe, ZHAI Dong-qi, JIANG Yuan-song, SUN Jian-hua, LI Jia-qi

(Wuhan National Laboratory for Optoelectronics of Huazhong Institute of Electro-Optics, Wuhan 430223, China)

**Abstract:** When traditional display and control systems have high-definition video display requirements, they often rely on industrial computers with dedicated graphics processing hardware such as GPUs and graphics cards, which has disadvantages such as large size and high power consumption. In response to these problems, we have studied the superposition display technology of high-definition video and graphical interface, which is based on ZYNQ SoC, an ARM+FPGA heterogeneous chip. Integrated image acquisition driver, DRM display driver and dma-buf memory sharing driver, combined with Qt graphics library to generate human-computer interaction and control interface, an embedded Linux operating system was customized on ARM. Multiple display layers' transparency and overlay relationships were controlled by libdrm interface on application level. ARM is interconnected with FPGA through the AXI bus, and the control logic resources complete the work of large data volume such as image transmission, decoding, image and Qt graphical interface overlay and display, thereby reducing the load of the CPU and giving full play to the advantages of ZYNQ heterogeneous chips. This software and hardware collaborative solution of superimposing high-definition image and graphics has the advantages of flexibility, low power consumption, small size, and low cost compared with traditional display and control solutions.

**Key words:** display technology; DRM framework; multi-layer; overlay; high-definition image

## 0 引言

高清视频与图形控制界面的叠加显示技术在很多领域有所应用<sup>[1]</sup>,如直升机的视频监控中,不仅要求实时显示侦察目标的状况,并且需要同时显示直升机在飞行过程中的速度、位置、高度等信息,并提供人机操作控制<sup>[2]</sup>,此时就需要视频图形叠加显示技术。此外,在舰船、海军光电、汽车等领域的显控终端也需要此项技术<sup>[3]</sup>。

传统的显示与控制系统在有高清视频显示需求时,往往采用专用的图形处理硬件如显卡、GPU 或者专用于视频显示的芯片满足高清视频显示的需求<sup>[4]</sup>,显示与控制软件通常部署在工控机中,在桌面操作系统环境下开发和运行人机交互界面。

ZYNQ 是 Xilinx 公司推出的一款可扩展处理平台,其结合了处理器系统 (Processing System) 以及可编程逻辑 (Programmable Logic) 两部分,是一款 ARM+

收稿日期: 2021-05-08

修回日期: 2021-09-12

基金项目: 国防科技预研项目 (302010408)

作者简介: 连成哲 (1994-), 男, 硕士, 助理工程师, 研究方向为计算机集成制造系统。

FPGA 的异构芯片,通过 AXI 数据交互协议进行片内处理器与可编程逻辑的高速互联<sup>[5]</sup>,PS 部分支持 Linux、VxWorks、 $\mu$ Cos 等多种嵌入式操作系统,可以很方便地使用图形库提供人机交互界面,同时通过 PL 部分对图像、视频等进行处理、显示。基于 ZYNQ SoC 的高清视频与图形界面叠加方案相较于传统方式,将图形采集、人机交互、视频与图形界面叠加、高清视频显示等功能集成在一个芯片上,具有灵活、低功耗、体积小等优势。

基于 ZYNQ SoC 平台,该文研究了高清视频与图形界面叠加的关键技术,包括 Video DMA 视频直接内存访问技术、DRM 显示子系统技术、DMA-buf 内存共享技术等。通过 ZYNQ 的 PS 端 Linux 系统下基于 Qt 产生图形界面,在 PL 端进行高清图像的采集、与 Qt 图形界面进行叠加并通过 PL 端 HDMI 接口进行显示输出,采用 DRM 显示框架对显示设备进行驱动和控制,进行同屏图层切换、多屏互动。

## 1 基于 ZYNQ SoC 的高清视频与图形界面叠加显示技术原理

在多个领域都有视频监控的需求,如海军光电显控终端,需要实时采集摄像头传感器的图像数据进行显示,并提供操控界面供用户使用,同时,需要在实时

图像上叠加波门、目标信息等图形和文字<sup>[6]</sup>,这需要在保证图像流畅度和清晰度的基础上进行多个显示图层的叠加。

基于 ZYNQ SoC 的高清视频与图形界面叠加显示技术的原理如图 1 所示。在数据流中,由 PL 端资源 VDMA 进行高清图像的实时采集,并缓存在 PS 端的 DDR 中;同时,由 Qt 图形界面库生成图像上的叠加图形与文字,以及鼠标、按钮等人机交互图形界面,通过帧缓存(FrameBuffer)对图形帧数据进行存储。在显示阶段,通过 PL 端的两路 VDMA 分别对图形和图像数据进行搬运,并在 PL 端的 OSD (Video on Screen Display) IP 核进行硬件叠加,通过编/解码 IP 进行 HDMI 编码后显示输出。而在 PS 端运行集成了 VDMA 控制驱动 VDMA Driver、内存零拷贝驱动 DMA-buf、显示设备管理驱动 DRM 以及图形库 Qt 的嵌入式 Linux 操作系统,并通过 libdrm 应用接口库对多个图层进行管理。该方案中,高清图像的采集、图像和图形数据的输出以及多图层叠加和显示编码都通过 ZYNQ 的可编程逻辑部分实现,ARM 处理器仅需对 PL 端初始化与配置即可实现高清图像与人机交互界面的显示,大大减轻了 CPU 的负载,提高视频的显示帧率与人机交互界面的流畅度。

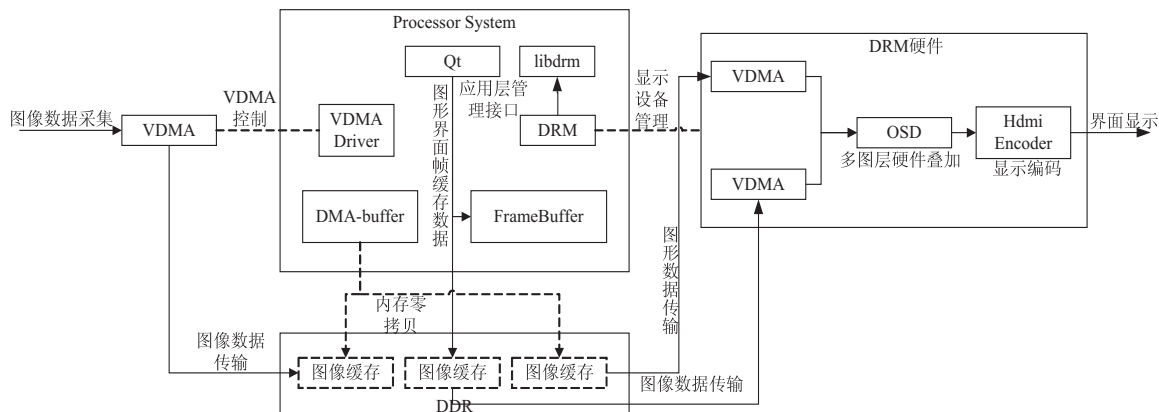


图 1 叠加显示技术原理

## 2 基于 SoC 的显示控制相关技术研究

### 2.1 Video DMA 视频直接内存访问

AXI Video Direct Memory Access (VDMA) 是一个 IP 软核,提供了存储器与 AXI-Stream 视频类型目标外围设备之间的高带宽的直接内存访问。许多视频应用程序都需要帧缓冲区来处理帧速率变更或图像缩放和裁剪,AXI VDMA 旨在允许视频接口和 AXI 接口之间进行高效的高带宽访问<sup>[7]</sup>。

其结构如图 2 所示,VDMA 分为 S2MM 写通道和 MM2S 读通道,分别表示将 stream 视频流数据通过

AXI 总线写入内存和将视频数据通过 AXI 总线从内存中读出,而不需要 CPU 过多的参与<sup>[8]</sup>,从而降低 CPU 的开销。

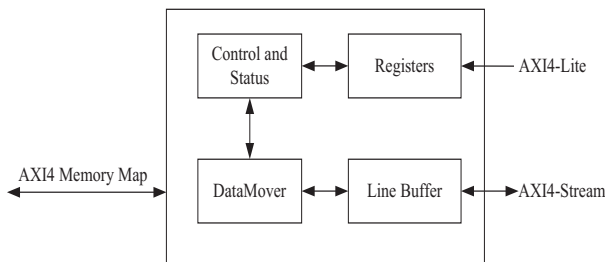


图 2 VDMA 结构框图

## 2.2 Direct Rendering Manager 显示架构

DRM 是 Linux 下的图形渲染架构,是一种对 GPU 显卡等显示设备的管理和封装技术。DRM 驱动架构主要包括以下几个元素:画布(FrameBuffer)、显示控制器(CRTC)、编码器(ENCODER)、连接器(CONNECTOR)和平面(PLANES)<sup>[9]</sup>。其关系如图 3 所示。

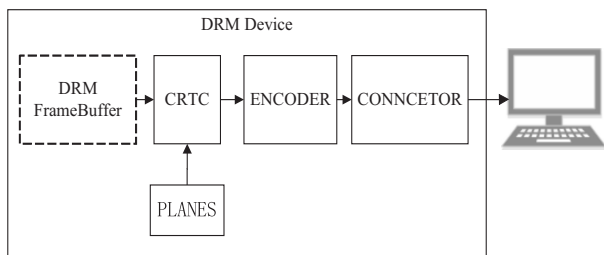


图 3 DRM 驱动架构

### (1) DRM FrameBuffer。

FrameBuffer 是一片用户程序和驱动都能访问的内存的抽象定义,表示了单个图层显示的内容,是与硬件无关的基本元素。DRM 显示架构与传统的 FrameBuffer 显示架构在 Linux 系统正常加载后都会会在设备节点下生成/dev/fb\* 节点,从而为用户应用提供接口,Linux 常用的图形界面库如 GTK、Qt 等都是基于

FrameBuffer 接口实现的<sup>[10]</sup>。

### (2) CRTC。

CRTC 是 CRT Controller 的缩写,是扫描输出缓冲的抽象表示,负责读取帧缓冲中的数据。CRTC 可以连接多个编码器(ENCODER),用于复制屏幕,并管理各种显示时序参数,是显示器配置的中心环节。

### (3) ENCODER。

ENCODER 将从 CRTC 中取得的像素数据转换为目标 CONNECTOR 支持的格式。

### (4) CONNECTOR。

CONNECTOR 是与物理显示设备的连接器,如 HDMI、DisplayPort、DSI 总线。

### (5) PLANES。

PLANE 代表一个可合成扫描输出缓冲的图像源,也称为硬件图层,用以实现 Overlay 的应用。

## 2.3 DMA-buf 内存共享技术

高清图像的采集与显示需要将系统中采集到的图像缓存拷贝至显示设备的缓存 buffer 中,当针对高清图像这种大数据量的应用来说,简单地依靠 memcpy 会消耗 CPU 大量的负载,DMA-buf 为不同设备、子系统之间进行内存共享提供了统一的机制<sup>[11]</sup>。典型的 DMA-buf 应用框图如图 4 所示。

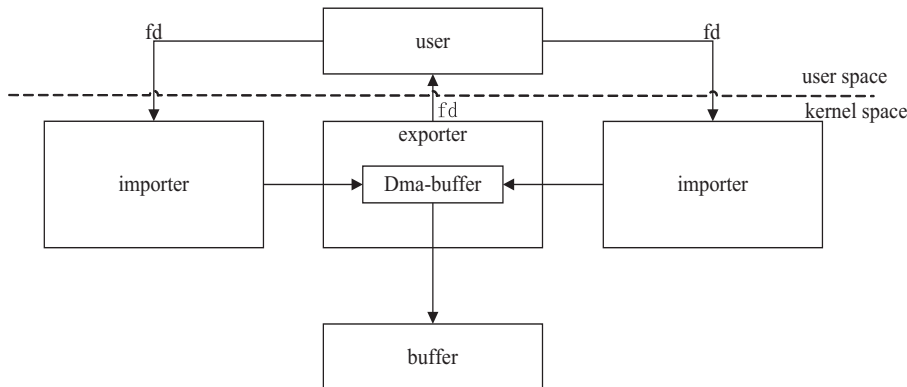


图 4 DMA-buf 应用框图

DMA-buf 框架下主要有两个角色对象,一个是 exporter,相当于 buffer 缓存的生产者,对应的是 importer,即 buffer 的使用者。各主要功能如下:

#### Exporter:

实现 DMA-buf 驱动框架中缓存管理回调函数;

允许其他使用者通过 DMA-buf 的 API 来使用共享缓存;

管理共享缓存的分配、包装等细节工作;

决策共享缓存的实际后端内存的来源;

管理 scatterlist 的迁移工作。

#### Importer:

是共享缓存的使用者之一;

访问共享缓存的 scatterlist,并提供将共享缓存映

射到自己地址空间的机制。

## 3 基于 ZYNQ SoC 的光电系统显示控制平台设计与实现

### 3.1 ZYNQ SoC 程序设计

在 ZYNQ SoC 中,高清图像采集、编码、图形图像叠加与显示在 PL 端具体实现,PS 端则通过图形库生成人机交互界面与叠加图形和文字,并通过 AXI 总线控制 PL 端软核对各功能进行配置。基于 ZYNQ SoC 的高清视频与图形界面叠加显示方案程序设计原理图如图 5 所示。

对图 5 所示程序中各功能模块做简单介绍。在高清图像采集中,由 Video in to AXI4-stream 的 IP 核将

视频图像信息转换为 AXI 总线类型的视频流,通过 AXI Video Direct Memory Access(VDMA0)写入至 ARM 端的缓存地址中。在图形界面生成模块中,由 ARM 中 Qt 图形界面库生成对应的人机交互图形界面和图像的叠加图形与文字,也存储在缓存中。在 DRM 显示子

系统中,由两路 VDMA 组成 DRM 系统的多层 PLANES,分别从对应缓存区中读取图形和图像数据,并在 Video on Screen Display 的 IP 核中进行硬件叠加,并通过 HDMI 或 DVI 接口输出显示。

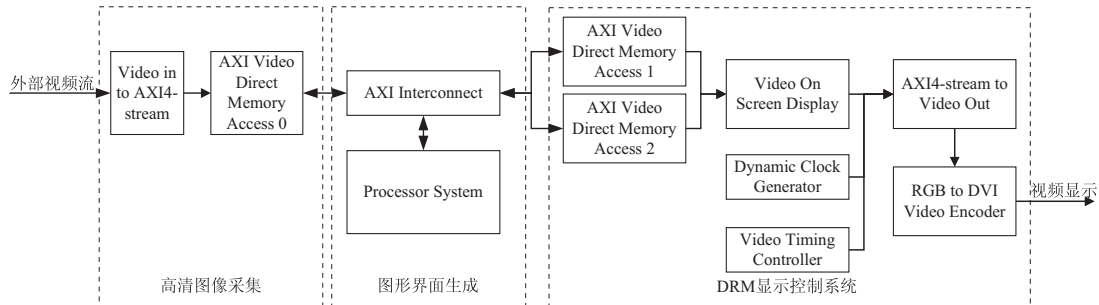


图5 ZYNQ SoC 程序设计

### 3.2 图像采集驱动

图像采集驱动主要通过控制 VDMA 将视频流数据写入缓存中,驱动流程如图6所示。

图像采集驱动对 VDMA 的操作在 file\_operation 结构体中实现,通过 axidma\_ioctl 函数对 VDMA 支持的配置如表1所示。

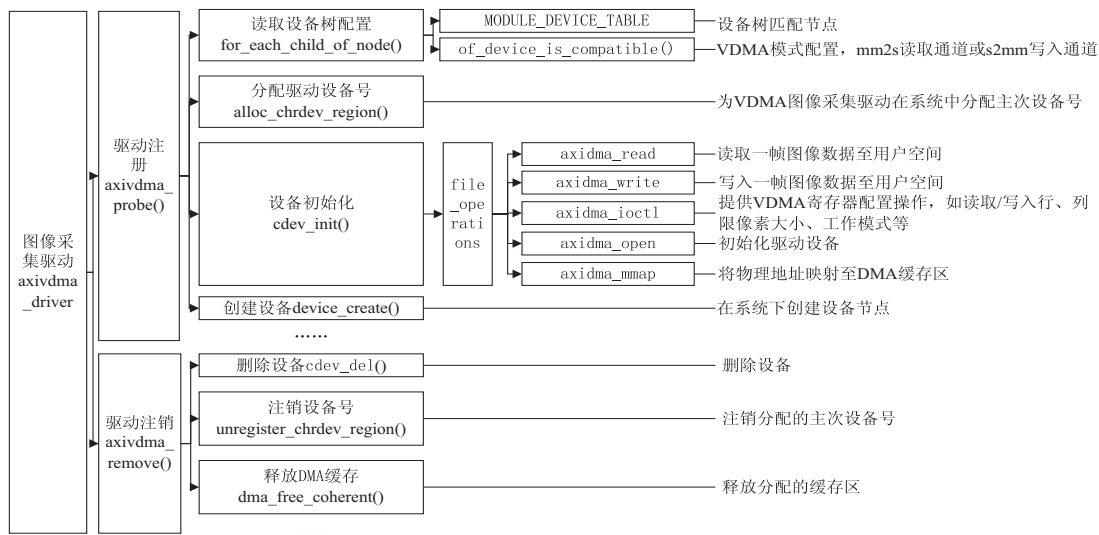


图6 图像采集驱动框架

表1 图像采集驱动支持的功能

参数	功能
RD_S2MM_VSIZE/RD_S2MM_HSIZE/RD_S2MM_FRAME_STRIDE	读取 S2MM 通道行/列/跨度
WR_S2MM_VSIZE/WR_S2MM_HSIZE/WR_S2MM_FRAME_STRIDE	设置 S2MM 通道行/列/跨度
START_MM2S_TX/STOP_AXIVDMA_TX	开始/停止读出图像
START_S2MM_RX/STOP_AXIVDMA_RX	开始/停止采集图像
RD_MM2S_VSIZE/RD_MM2S_HSIZE/RD_MM2S_FRAME_STRIDE	读取 MM2S 通道行/列/跨度
WR_MM2S_VSIZE/WR_MM2S_HSIZE/WR_MM2S_FRAME_STRIDE	设置 MM2S 通道行/列/跨度
WR_MM2S_WROKMODE/RD_MM2S_WROKMODE	设置/读取 MM2S 通道工作模式
WR_S2MM_WROKMODE/RD_S2MM_WROKMODE	设置/读取 S2MM 通道工作模式
READ_BUFFER/WRITE_BUFFER	读取/写入缓存

### 3.3 多层 DRM 配置

Xilinx 为 DRM 相关的硬件 IP 核提供了驱动,并集成在了 Linux 内核中,相关驱动代码在 Linux 内核目录

的 drivers/gpu/drm/xilinx 路径下。在 3.1 节 SoC 程序设计的基础上,需要配置 DRM 设备树,为 Linux 正常使用该设备提供支持。



DRM 的设备树配置如下,对应 2.2 节所述,两路 VDMA 对应 DRM 显示系统的 PLANES, xlnx, encoder-slave 节点对应 ENCODER, xlnx, connector-type 节点对应 CONNECTOR。

```
xilinx_drm {
    compatible = "xlnx,drm";
    xlnx,vtc = <&v_tc_0>;
    xlnx,connector-type = "HDMI";
    xlnx,osd = <&v_osd_0>;
    xlnx,encoder-slave = <&hdmi_encoder_0>;
    clocks = <&axi_dynclock_0>;
    planes {
        xlnx,pixel-format = "argb8888";
        plane0 {
            dmas = <&axi_vdma_1 0>;
            dma-names = "axi_vdma_1";
        };
        plane1 {
            dmas = <&axi_vdma_2 0>;
            dma-names = "axi_vdma_2";
        };
    };
};
```

### 3.4 嵌入式操作系统与运行环境移植

操作系统与运行环境移植主要包含以下步骤:

操作系统制作:

PetaLinux 工具提供在 Xilinx 处理系统上定制、构建和调配嵌入式 Linux 解决方案所需的所有组件,简

化 Linux 产品的开发<sup>[12]</sup>。在 Linux 内核中将 3.2 节所述的图像采集驱动添加进内核并修改对应的配置文件,同时修改 system-user.dtsi 设备树文件,对图像采集驱动与 DRM 显示系统进行配置后,经过编译生成对应的系统镜像。

Qt 图形库移植:

Qt 是一个开源的跨平台 C++ 图形界面应用程序框架<sup>[13]</sup>,源码可在官网获取,通过赛灵思提供的交叉编译链进行交叉编译后,将对应的库文件拷贝至嵌入式 Linux 的文件系统中,并配置 Qt 库的环境变量从而支持 Qt 程序的运行。

libdrm 库移植:

libdrm 库是 DRM 系统的应用层封装,提供图层配置、显示链路配置、显存管理等,通过 libdrm,用户可以间接调用 DRM 系统<sup>[14]</sup>。libdrm 源码可在 <https://dri.freedesktop.org/libdrm/> 下载,交叉编译后存储在 linux 的文件系统中,并配置环境变量从而支持 libdrm 程序的运行。

### 3.5 测试用例设计

以光电跟踪设备为例,在光电跟踪设备中,通常有图像传感器接收的真实图像数据并在图像上叠加对应的目标识别信息、波门、目标状态等信息,并提供设备操控界面<sup>[15]</sup>。在上述设计搭建的显示方案基础上,测试用例通过 Qt 生成图形数据,与实时采集的高清图像数据进行叠加显示,并动态的配置多个图层的显示关系,程序设计流程如图 7 所示。

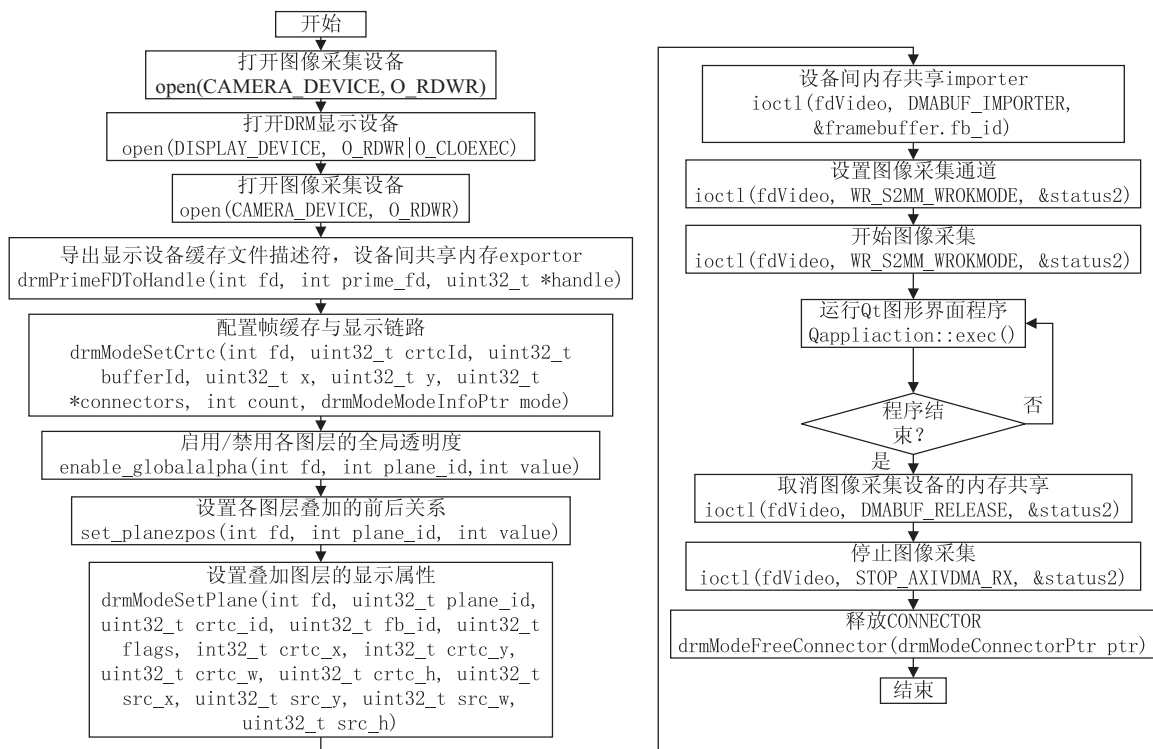


图 7 测试程序设计流程

测试程序将采集的图像层作为 DRM 子系统的 plane1, Qt 产生的图形界面作为 plane0, 同时配置 plane0 作为 Overlay 叠加图层, 并禁用了该图层的全局透明度, 从而支持各个像素点透明度的配置, 在 Qt 中将需要显示视频图像的像素部分 Alpha 通道设置为 0。

程序运行现场如图 8 所示。其中, 屏幕上半部分 1 920 \* 512 像素位置底图为实时仿真海面图像数据, 标尺与刻度、波门以及状态信息文字等由 Qt 产生, 并将 1 920 \* 512 位置背景像素的透明度置为 0, 下半部分同样由 Qt 产生的人机操控界面, 背景不透明。

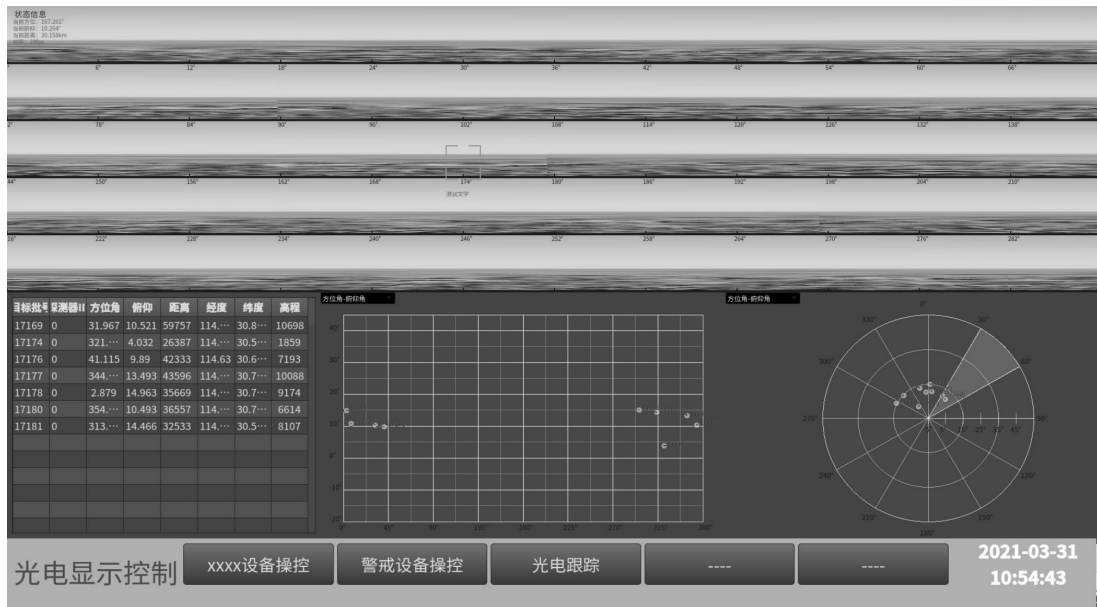


图 8 程序运行现场图

## 4 结束语

基于 ZYNQ SoC 提出了一种高清视频与图形界面叠加的显示技术。通过 dma-buf 机制共享图像采集设备与 DRM 显示设备之间的图像帧缓存, 且图像的采集、图像缓存区至硬件显示接口的数据传输、图像编码都通过 SoC 的 PL 端完成, 不需要 CPU 过多的参与, 大大减少了 CPU 的负担, 提高了高清图像的显示速率。通过实测, 在显示同一分辨率的图像下, 通过该叠加方案显示时与通过 CPU 直接叠加显示时 CPU 占用率从 89% 降至 13%, 帧率从 11 fps 提升至 30 fps 左右, 满足常用的显示需求。

### 参考文献:

- [1] 林春富. 基于 DRM 子系统的汽车中控系统显示控制解决方案的研究[D]. 厦门: 厦门大学, 2017.
- [2] ANONYMOUS. Army Advances manned-unmanned systems teaming[J]. Signals, 2014, 69(4): 11.
- [3] 李杉格, 林静, 李杰. 一种机载光电吊舱显控软件设计与实现[J]. 软件导刊, 2018(8): 153-156.
- [4] 朱方. 基于 MPSoC 的移动视频监控关键技术研究[D]. 南京: 东南大学, 2016.
- [5] YU Minwu. Film and television culture dissemination based on ZYNQ embedded digital image processing[J]. Microprocessors and Microsystems, 2021, 82: 103921.
- [6] 安东. 国外海军最新光电系统综述[J]. 现代军事, 2016(10): 73-77.
- [7] 吴汶泰, 詹璨铭. 基于 Zynq 的 AXI 总线数据传输软件优化[J]. 通信技术, 2017, 50(7): 1576-1580.
- [8] GAO Xi, ZHANG Liyan. The research and implementation of sobel calculatory based on FPGA[J]. World Scientific Research Journal, 2021, 7(4): 8.
- [9] 褚享强. 基于 ZYNQ 的 HDMI 显示端设计与实现[J]. 电视技术, 2017, 41(2): 23-27.
- [10] OH H W, KIM J K, HWANG G B. The design of a 2D graphics accelerator for embedded systems[J]. Electronics, 2021, 10(4): 469.
- [11] 陈利锋, 姚秀文. 基于视频流的嵌入式 GUI 系统设计与实现[J]. 计算机工程, 2014, 40(12): 267-271.
- [12] 孙跃祥, 郭锐锋, 尹震宇, 等. 基于 Zynq 平台的 EtherCAT 主站系统设计与实现[J]. 组合机床与自动化加工技术, 2020(11): 112-115.
- [13] 杨柳, 岳坤, 庞和明, 等. Qt/Embedded 及嵌入式 Linux 在智能监控系统控制中的应用[J]. 计算机应用, 2010, 30(S1): 289-291.
- [14] 刘桦杰. 基于 Xilinx Zynq-7000 芯片的成像开发平台的研究与实现[D]. 杭州: 浙江大学, 2018.
- [15] ZABELS R, OSMANIS K, NARELS M. AR displays: next-generation technologies to solve the vergence - accommodation conflict[J]. Applied Sciences, 2019, 9(15): 77-79.