

## 基于 FPGA 的机载视频图形显示系统架构设计与优化

随着航空**电子技术**的不断发展,现代机载视频图形显示系统对于实时性等性能的要求日益提高。常见的系统架构主要分为三种:

(1) 基于 GSP+VRAM+ASIC 的架构,优点是图形 ASIC 能够有效提高图形显示质量和速度,缺点是国内复杂 ASIC 设计成本极高以及工艺还不成熟。

(2) 基于 DSP+FPGA 的架构,优点是,充分发挥 DSP 对算法分析处理和 FPGA 对数据流并行执行的独特优势,提高图形处理的性能;缺点是,上层 CPU 端将 OpenGL 绘图函数封装后发给 DSP, DSP 拆分后再调用 FPGA,系统的集成度不高,接口设计复杂。

(3) 基于 FPGA 的 SOPC 架构,优点是,集成度非常高;缺点是逻辑与 CPU 整合到一起,不利于开发。

经过对比,机载视频图形显示系统的架构设计具有优化空间,值得进一步的深入研究,从而设计出实时性更高的方案。

本文设计一种基于 FPGA 的图形生成与视频处理系统,能够实现 2D 图形和字符的绘制,构成各种飞行参数画面,同时叠加外景视频图像。在保证显示质量的同时,对其进行优化,进一步提高实时性、减少内部 BRAM 的使用、降低 DDR3 的吞吐量。

### 1 总体架构设计

本系统总体设计方案如图 1 所示。以 Xilinx 的 Kintex-7FPGA 为核心,构建出一个实时性高的机载视频图形显示系统。上层 CPU 接收来自飞控、导航等系统的图形和视频控制命令,对数据进行格式化和预处理后,通过 PCIe 接口传送给 FPGA。本文主要是进行 FPGA 内部逻辑模块的设计和优化。

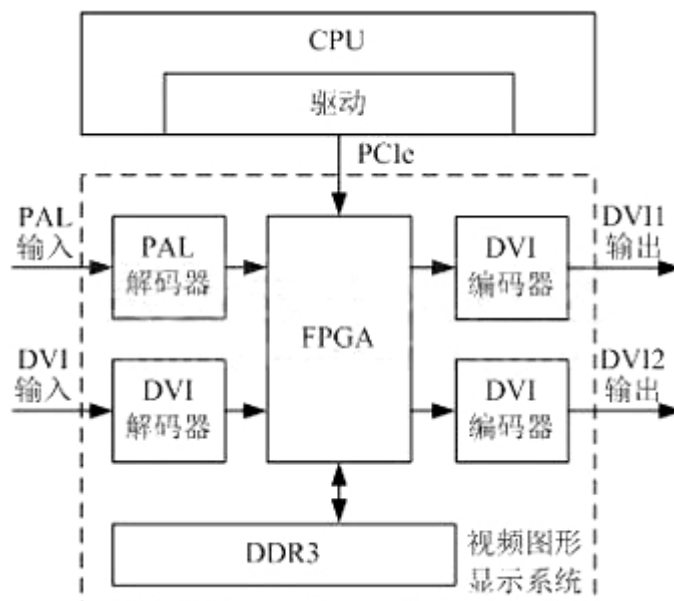


图 1 机载显示系统总体设计框图

## 2 机载显示系统架构设计

机载显示系统设计主要包括 2D 绘图、视频处理和叠加输出。2D 绘图功能包括直线、圆、字符等的快速生成。视频处理功能包括输入视频选择、视频缩放、旋转、翻转等处理。叠加输出功能，将视频作为背景与图形叠加，送到两路 DVI 输出，一路经过预畸变校正后输出到平显上，另一路直接输出来进行地面记录。

为了满足上述功能，FPGA 逻辑设计的整体流程图如图 2 所示。

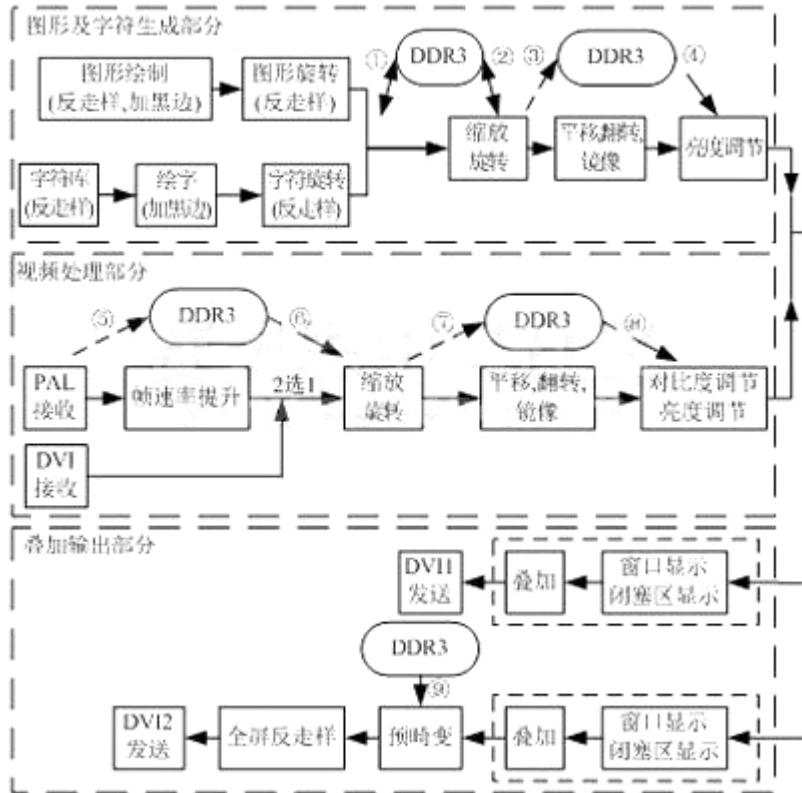


图 2 FPGA 逻辑设计的整体流程图

### 2.1 实时性分析

视频处理既要实现单纯的外视频处理，同时能够实现叠加后视频处理。以旋转处理为例，若在单纯外视频旋转处理后，与图形叠加，再进行叠加后旋转处理，延迟非常大。因此为了提高实时性，考虑将图形整体和外视频分别进行旋转处理后，再相互叠加。整个流程中，帧速率提升模块延迟最大。

#### 2.1.1 帧速率提升算法

帧速率提升指在原有的图像帧之间插值出新的图像帧。常见的帧速率提升算法主要包括帧复制法、帧平均法和运动补偿法。综合考虑显示效果和实时性要求，最终选择帧复制法。帧复制法易于实现、计算量低。其表达式为：

$$f_1(\bar{x}, T) = f_0(\bar{x}, T - 1) \quad (1)$$

此处输入 PAL 视频帧速率为 25 帧/秒，输出 DVI 视频帧速率为 60 帧/秒，即在 0.2s 内将 5 帧图像插值到 12 帧。如图 3 所示，DDR3 中开辟 5 帧存储空间用于存放 25Hz 的原始图像，在 0.2 内输入 5 帧原始图像，输出 12 帧图像。延迟为 PAL 的 1.5~2.6 帧，最大延迟为： $2.6 * (1 / 25Hz) = 104ms$ 。

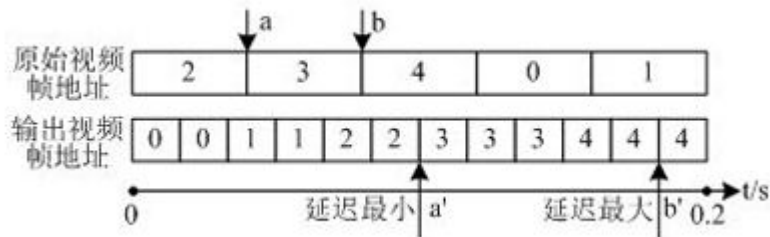


图 3 帧速率提升示意图

## 2.2 BRAM 资源占用

本文设计的机载显示系统利用一片 DDR3 作为外部存储器，所有图形和视频数据都需要缓存到 DDR3 中。为了解决数据存储冲突，需要将数据先缓存到内部 BRAM 中。XC7k410T 共有 795 个 36Kb 的 BRAM。整个流程中，BRAM 资源占用最大的是图形整体旋转和视频旋转模块。

### 2.2.1 视频旋转算法

视频旋转包括两个基本操作<sup>[5]</sup>：空间坐标变换和灰度级插值。灰度级插值算法选择双线性插值，此处重点讨论空间坐标变换的选择。空间坐标变换主要包括两种：正向映射和反向映射。由于正向映射旋转后得到浮点坐标，而图像坐标是整点，使得旋转图像存在“空洞”现象。因此采用反向映射。反向旋转映射算法思想是：对旋转后的每行每像素坐标  $(x', y')$  进行遍历，绕屏幕中心  $(x_0, y_0)$  沿逆时针旋转  $-\theta$  角度后，得到旋转前图像坐标为  $(x, y)$ 。

$(x', y')$  与  $(x, y)$  之间的反变换公式为：

$$\begin{cases} x = (x' - x_0) \cos \theta + (y' - y_0) \sin \theta + x_0 \\ y = -(x' - x_0) \sin \theta + (y' - y_0) \cos \theta + y_0 \end{cases} \quad (2)$$

反向旋转映射优点是，旋转后坐标反向旋转，除了超出原始坐标范围的，在旋转前坐标中都能对应到浮点坐标，并可以用该坐标邻域的像素点来唯一确定该坐标的像素值，不会出现“空洞”现象。

充分考虑项目的特殊情况，由于项目中旋转是用来校正屏幕的，旋转角度范围是  $-10 \leq \theta \leq 10$ 。如图 4 所示，对旋转后目标图像的第 N 行进行反向旋转获取坐标时，原始坐标在 N-M~N+M 行之间。分辨率为  $1920 \times 1080$  时，角度为  $10^\circ$  时，M 为  $1920 / 2 * \sin(10 * \pi / 180) = 167$  行，即当计算输出第 N 行时，需要知道原始图像的 N-167~N+167 行来获取，即需要缓存 334 行，每行需要  $1920 * 16 = 30Kb$ ，即一共需要 279 个 36Kb 的 BRAM。

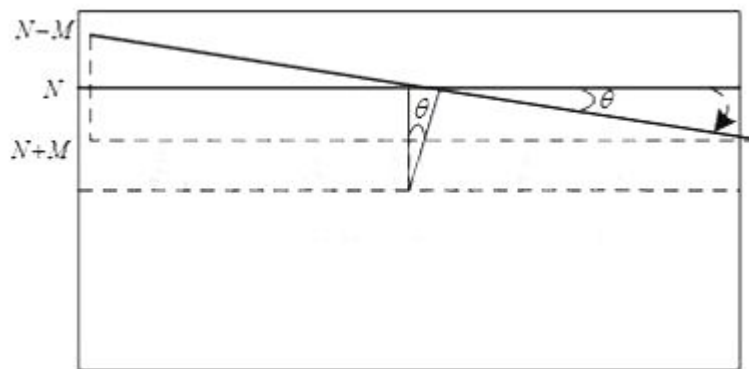


图 4 视频旋转算法示意图

### 2.3 DDR3 吞吐量分析

本系统处理的数据量大，FPGA 内部的存储资源无法满足数据存储要求，需要配置系统外部存储器 DDR3。从图 2 可以看出，整个系统流程最多经过 DDR3 共 9 次，下面依次介绍每次读写 DDR3 的必要性和数据量。

①图形及字符生成模块读写数据，由于图形及字符生成时，没有严格按照屏幕自上而下、自左而右的顺序，所以每一帧图形都需要存入到 DDR3 中，并等待一帧处理完再进行整体的其他处理。由于像素点操作会涉及到读取背景值，所以是双向的。考虑吞吐量最大的情况，即图形生成模块对每帧图形的一半像素点都进行一次读写操作，则图形生成模块读写数据量为

$$60\text{Hz} * (1920 * 1080) * 16\text{bit} * 2 / 2 = 237.3\text{MB} / \text{s} \quad (3)$$

②图形缩放模块读数据及清屏操作，由于缩放模块是对整个画面的处理，所以需要从 DDR3 中读取出来。而由于图形及字符不是对每个像素点都进行操作的，所以取出后，需要进行清屏操作。则读写数据量为 474.6。

③图形旋转模块写数据，由于图形处理速度和视频处理速度不完全匹配，所以先将缩放及旋转后的数据存储到 DDR3，等待视频处理完后，再统一取出，写数据量为 237.3MB/s。



④图形输出读数据,同时进行平移、翻转、镜像等坐标变换操作,写数据量为 237.3MB/s。

⑤PAL 视频输入数据,为了实现去隔行和帧速率转换,必须将数据存储到 DDR3 中进行变换操作,读写数据量为

$$25Hz * (720 * 576) * 16bit = 19.8MB / s \quad (4)$$

⑥视频缩放模块读数据,从 DDR3 中取出进行缩放操作,读数据量为

$$60Hz * (720 * 576) * 16bit = 47.5MB / s \quad (5)$$

⑦视频旋转模块写数据,由于视频处理速度和图形处理速度不完全匹配,所以先将缩放及旋转后的数据存储到 DDR3,等待图形处理完后,再统一取出,读写数据量为 237.3MB/s。

⑧视频输出读数据,同时进行平移、翻转、镜像等坐标变换操作,读写数据量为 237.3MB/s。

⑨预畸变参数读数据,用 64 位来存储每个像素点对应的四个预畸变参数,则读数据量为

$$60Hz * (1920 * 1080) * 64bit = 949.2MB / s \quad (6)$$

表 1 为该系统数据吞吐量的计算表,其吞吐量合计为 2677.6MB/s。

	帧频率 (Hz)	分辨率	数据宽度 (bit)	吞吐量 (MB/s)
图形生成读写	60	1920×1080	16*2/2	237.3
图形缩放读及 清屏操作	60	1920×1080	16*2	476.4
图形旋转写	60	1920×1080	16	237.3
图形输出读	60	1920×1080	16	237.3
PAL 写	25	720×576	16	19.8
视频缩放读	60	720×576	16	47.5
视频旋转写	60	1920×1080	16	237.3
视频输出读	60	1920×1080	16	237.3
预畸变读数	60	1920×1080	64	949.2

表 1 系统数据吞吐量计算表

本文采用 DDR3 作为系统外部存储器,其型号为 W3H128M72E,数据宽度为 72 比特(64 比特为数据位,8 比特为校正位),采用的时钟为 400MHz,由于 DDR3 在上升沿和下降沿都进行数据的读写操作,等效于其内部读写时钟为 800MHz,即数据带宽为 6400MB/s(800MHz\*64bit),满足本文设计系统的数据吞吐量要求。

### 3 机载显示系统架构优化

设计的机载显示系统架构能够满足性能要求，但是还需要进一步优化。如图 5 所示，改变不同模块之间的顺序来优化设计，同时改进算法。具体改变如下：

①图形整体相对于屏幕的缩放和旋转功能在 CPU 端发送命令前实现，因为 CPU 端旋转和缩放是针对顶点进行的，方便快捷，同时减少了 FPGA 的 BRAM 资源占用，减少了进出 DDR3 的次数；

②改进帧速率提升算法，进一步减少延迟，提高实时性；

③改进视频旋转算法，进一步降低缓存区的大小，减少 BRAM 的占用率；

④帧速率提升和平移、翻转、镜像都需要通过读写 DDR3 来完成，将两者合并，同时完成，减少进出 DDR3 的次数。

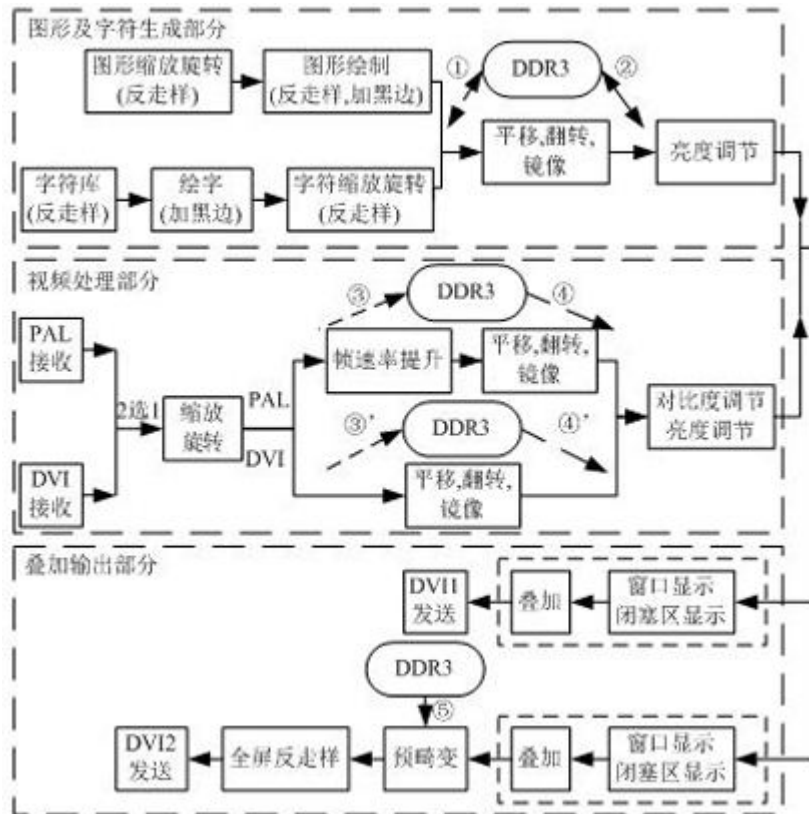


图 5FPGA 逻辑优化的整体流程图

#### 3.1 实时性分析

实时性是机载显示系统重要的衡量标准之一，为了确保飞机运行安全，必须确保视频处理的各个模块都有较高的实时性。视频采集、视频缩放、视频校正、视频输出延迟都是几行，延迟时间在以内。帧速率提升模块的延迟远大于其他各个模块延迟之和，需要进一步改进，在保证显示质量的同时，进一步降低延迟时间。

##### 3.1.1 帧速率提升算法优化

改进的帧速率提升算法仍使用帧复制法。在 DDR3 中开辟 4 个存储空间做切换用于存放帧速率为 25Hz、场速率为 50Hz 的 PAL 图像。有 4 个场缓存区，当接收当前帧的奇场后与前一帧的偶场结合成一帧数据输出。

帧速率改进算法示意图如图 6 所示。A 场正好写完，B 场正好读完，下一帧读取 A 场数据，这样延迟为 PAL 的 1 场（半帧）；A 场正好还差 1 行写完，B 场已读完，下一帧继续读 B 场，这样延迟为 PAL 的  $1 + (25/60) = 1.42$  场。延迟为 PAL 的 1~1.42 场。最大延迟为  $1.42 * (1 / 50Hz) = 28.4ms$ 。

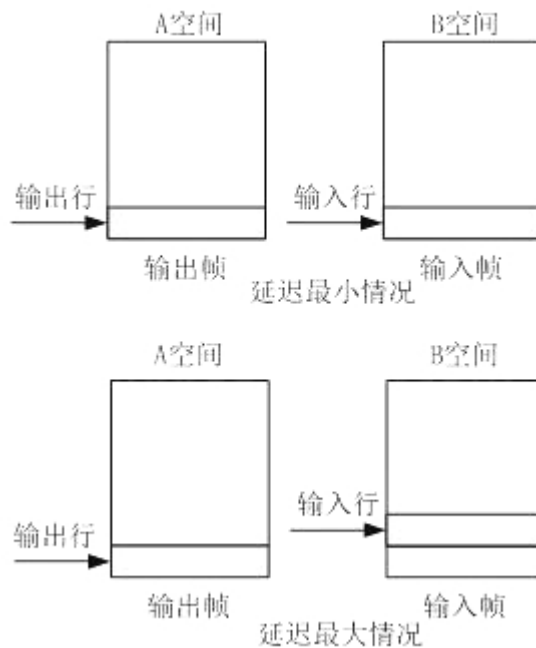


图 6 帧速率改进算法示意图

### 3.2 BRAM 资源占用

原设计的机载显示系统架构使用反向映射的方法实现旋转算法，每一行旋转后数据反向旋转时需要缓存 334 行视频旋转前数据，即需要 279 个 36Kb 的 BRAM。相对于其它模块缓存几行相比，占用了大量的 BRAM 空间，因此需要改进。

#### 3.2.1 视频旋转算法优化

视频旋转提出了一种改进的旋转映射法，降低缓存空间。示意图如图 7 所示。对以行扫描的方式获取的视频图像，缓存两行就能开始旋转处理，先进行正向映射，根据当前两行对应的旋转后浮点坐标，找到两行内的整点坐标，再对其进行反向映射，利用当前两行来得到旋转后整点坐标的像素值。

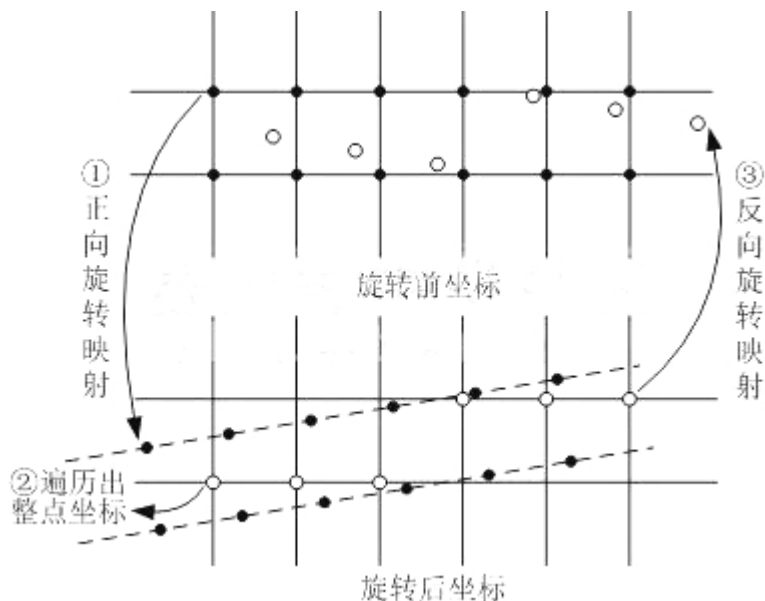


图 7 视频旋转改进算法示意图

该算法涉及原始图像中的  $2 \times 2$  大小邻域，为了提高该模块的处理速度，设计了一组由三个双端口块存储器 BRAM 组成的原始图像数据缓存器。每个 BRAM 用来存储 1 行原始图像的数据，3 个 BRAM 中存储的原始图像数据包括当前旋转计算涉及的两行原始图像数据以及下一行旋转计算涉及的一行原始图像数据。因此，需要缓存 3 行，使用 3 个 36Kb 的 BRAM。

### 3.3 DDR3 吞吐量分析

从图 5 可以看出，优化后的系统流程最多经过 DDR3 共 5 次，下面依次介绍每次读写 DDR3 的必要性和数据量。

①图形及字符生成模块读写数据，此与原模块相同，则图形生成模块读写数据量为

$$60\text{Hz} * (1920 * 1080) * 16\text{bit} * 2 / 2 = 237.3\text{MB} / \text{s} \quad (7)$$

②图形输出模块读数据及清屏操作写数据，由于视频输出模块是对整个画面的处理，所以需要从 DDR3 中读取出来，而由于图形及字符不是对每个像素点都进行操作的，所以取出后，需要进行清屏操作。则读写数据量为 474.6MB/s。

③视频旋转写数据，旋转后的数据没有严格按照屏幕自上而下的顺序，必须将数据存储到 DDR3 中进行变换操作，读写数据量为

$$60\text{Hz} * (1920 * 1080) * 16\text{bit} = 237.3\text{MB} / \text{s} \quad (8)$$

④视频输出模块读数据，从 DDR3 中取出进行视频输出操作，取出的同时还能进行一些坐标变换操作，如平移、翻转、镜像等，读写数据量为

$$60\text{Hz} * (1920 * 1080) * 16\text{bit} = 237.3\text{MB} / \text{s} \quad (9)$$

⑤预畸变参数读数据，用 64 位来存储每个像素点对应的四个预畸变参数，则读写数据量为



$$60\text{Hz} * (1920 * 1080) * 64\text{bit} = 949.2\text{MB} / \text{s} \quad (10)$$

表 2 为该系统数据吞吐量的计算表，其吞吐量合计为 2135.7MB/s。DDR3 的数据带宽为 6400MB/s（800MHz\*64bit），满足本文设计系统的数据吞吐量要求。

	帧频率 (Hz)	分辨率	数据宽度 (bit)	吞吐量 (MB/s)
图形生成读写	60	1920×1080	16	237.3
图形输出读及 清屏写数	60	1920×1080	16*2	474.6
视频旋转写	60	1920×1080	16	237.3
视频输出读	60	1920×1080	16	237.3
预畸变读数	60	1920×1080	64	949.2

表 2 优化后系统数据吞吐量计算表

## 结论

本文设计一种基于 FPGA 的机载显示系统架构，能够实现 2D 图形绘制，构成各种飞行参数画面，同时叠加外景视频图像。实时性方面，帧速率提升模块延迟最大为；BRAM 资源占用方面，视频旋转算法需要 279 个 36Kb 的 BRAM；DDR3 吞吐量方面，系统吞吐量为 2677.6MB/s。

优化后的机载显示系统，实时性方面，帧速率提升模块延迟最大为；BRAM 资源占用方面，视频旋转算法需要 3 个 36Kb 的 BRAM；DDR3 吞吐量方面，吞吐量为 2135.7MB/s。

经过对比分析，优化后的机载显示系统实时性提高、BRAM 资源占用减少、吞吐量降低，整体性能得到了提升。