

文章编号: 2096-1618(2021)01-0062-06

# 基于 ZYNQ 的视频处理平台框架设计

万 垚<sup>1</sup>, 周 咏<sup>2</sup>

(1. 成都信息工程大学通信工程学院, 四川 成都 610225; 2. 成都工业学院电子工程学院, 四川 成都 611730)

**摘要:**随着机器视觉的发展,针对目前的嵌入式设备对视频数据采集速度慢、分辨率低以及实时性不高的问题,提出一种基于 Xilinx ZYNQ 芯片的实时视频采集系统,该芯片采用 ARM+FPGA 的异构设计,利用该特性搭建软硬件协同设计的视频处理平台。FPGA 端实现图像的预处理,ARM 端实现对外设的配置,图像传输则利用 AXI 总线通过 HP 端口将数据高速存储到 DDR 中,最后利用 HDMI 端口对视频进行显示。通过利用 FPGA 的并行性和芯片中的高性能总线在保证系统低功耗的同时相比于一般的嵌入式设备具有更快的处理速度和更大的带宽以及实时性。为分布式部署监控系统以及实现更复杂的图像处理算法提供可靠稳定的平台。

**关键词:**通信集成电路;嵌入式系统;SPI;IMX222;ZYNQ;AXI 总线;FIFO

**中图分类号:**TP391.41

**文献标志码:**A

**doi:**10.16836/j.cnki.jcui.2021.01.010

## 0 引言

近来无人机频繁闯祸,多次发生无人机干扰民航客机正常飞行的事件,严重影响了处于低空状态的民航客机的飞行安全。现有的应对措施对于有效遏制无人机闯入尤其是恶意闯入限制区域存在功耗高、效率低、实时性差等缺点。鉴于此,提出一种基于 Xilinx ZYNQ 芯片的实时视频采集系统,该芯片采用 ARM+FPGA 的异构平台,将两者集成在—款芯片中。在满足低功耗的同时相比传统的嵌入式设备对视频流采集的实时性更高。该系统实现在摄像头端对视频流进行采集、存储和显示,其低功耗的特性可分布式部署在机场、港口、监狱等限制区域,为监控低空不明飞行器提供稳定、可扩展的平台。

## 1 视频采集系统总体结构设计

系统设计分视频采集、视频传输存储和视频显示 3 个模块。系统的设计和验证采用依元素的 EAGLGO 板卡作为开发平台,该平台采用 ZYNQ 系列型号为 xc7z020clg400 的芯片作为主控芯片。主控芯片实现对视频的传输和存储。SONY 公司的 IMX222 图像传感器完成对视频流的采集。最后利用板卡自带的 HDMI 接口作为视频流输出。主控芯片 ZYNQ 由 PS 部分和 PL 部分组成,PS 部分包含 2 个 Cortex-A9 的双核 ARM,ARM 采用 SDK 裸机开发,利用其自带的 GPIO 口和 SPI 总线对 IMX222 图像传输器进行配置,激活摄像头。PL 部分为可编程逻辑部分,即为传统的 FPGA 部分。系统对图像数据的处理以及传输、存储和显示

均由 PL 实现。PL 和 PS 两部分的互联由 AXI(高级外设接口)实现,这种片内互联的形式大大降低了 ARM 和 FPGA 之间的通信开销,使数据传输实时性高<sup>[1]</sup>。系统总体结构设计如图 1 所示。

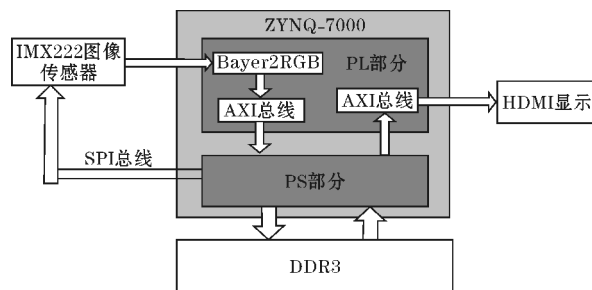


图1 系统总体结构设计

## 2 视频采集系统模块设计

### 2.1 摄像头采集模块设计

采用 IMX222 传感器进行视频流的采集输入,该传感器支持 720P 和 1080P 的视频模式。利用 PS 端的 SPI 总线对 IMX222 传感器进行配置。将摄像头的配置数据通过 SPI 总线写到图像传感器中<sup>[2]</sup>。SPI 的配置时序如图 2 所示。

该设计传入的视频流为 1080P 模式,图 3 为 1080P 模式下图像传感器需要配置的寄存器数据<sup>[3]</sup>。

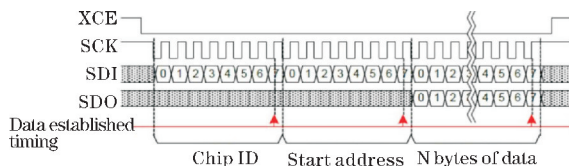


图2 SPI 配置时序图

Register details			Setting value				Function	
Register name	Address	Bit	Initial value	10 bit		12 bit		
				15 (frame/s)	30 (frame/s)	15 (frame/s)		30 (frame/s)
MODE	02h	[3:0]	00h	Fh				HD1080 p mode
HMAX	03h	[7:0]	044Ch	0898h	044Ch	0898h	044Ch	Horizontal (H) direction clock number designation
	04h	[5:0]						
	05h	[7:0]						
VMAX	06h	[7:0]	04E2h	04E9h				Vertical (V) direction line number designation
FRSEL		[2:0]		0h	1h	0h	1h	Output data rate designation
QPORTSEL	19h	[4:3]	1h	0h	0h	0h	0h	Parallel CMOS SDR output
				1h	1h	1h	1h	Parallel CMOS DDR output
				2h	N/A	2h	N/A	Serial LVDS 1ch output
				3h	3h	3h	3h	Serial LVDS 2ch output
M12BEN		[6:5]	0h	0h				Output gradation setting
ADRES	12h	[1]	0h	0h		1h		AD gradation setting
WINPH	14h	[7:0]	000h	000h				Designation of upper left coordinate for cropping position (Horizontal)
	15h	[3:0]						
WINPV	16h	[7:0]	000h	03Ch				Designation of upper left coordinate for cropping position (Vertical)
	17h	[3:0]						
WINWH	18h	[7:0]	7C0h	7C0h				Cropping size designation (Horizontal)
	19h	[3:0]						
WINWV	1Ah	[7:0]	4C9h	451h				Cropping size designation (Vertical)
	1Bh	[3:0]						
IOBITA	21h	[7]	0	1	0			Adjustments register for each operation mode.
720PMODE	22h	[7]	0	0				Sets in 720 p mode only.
IOBITB	7Ah	[7:0]	00h	40h		00h		Adjustments register for each operation mode.
IOBITC	7Bh	[7:0]	00h	02h		00h		
IOB1080 P	96h	[7:0]	226h	44Ch	226h			
	96h	[3:0]						
IOB1080 P	9Ah	[7:0]	44Ch	44Ch	226h			
	9Bh	[3:0]						
PRES	CEh	[6:0]	16h	16h				
DRES	CFh	[7:0]	082h	082h				
	DOh	[0]						

图 3 IMX222 1080P 模式寄存器

通过 SPI 总线配置完 IMX222 图像传感器后,需要对输出的嵌入式同步字和有效行场信号进行解析才能得到能够进行显示的图像数据,解析同步字以及产生行场同步信号均由 FPGA 实现。同步字输出时序图如图 4 所示。图像数据分为两种类型,无效行 (invalid line)

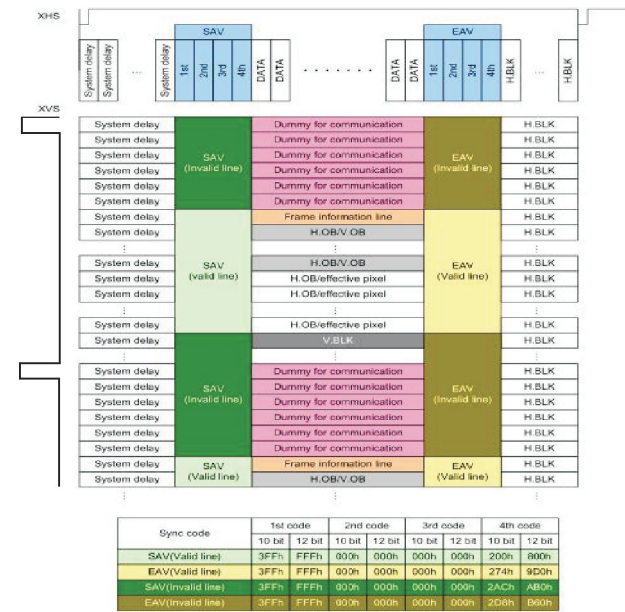


图 4 同步字输出时序图

表 1 同步字解析状态机流程图

当前状态	状态功能	状态转移条件	状态对外输出
状态 1	等待无效行	Invalid_flag = 1	状态 2
状态 2	等待有效行	valid_flag = 1	状态 3
状态 3	产生 vsync	EAV	状态 4
状态 4	计数 24 个无效区	计数完 24 行图像数据	状态 5
状态 5	启动水平和垂直方向计数器产生有效区域标志	完成 1080 行所有像素的检测	状态 1

和有效行 (valid line), 其中 SAV 和 EAV 分别表示图像数据一行的开始和结束。设计通过移位寄存器的方式检测 SAV 和 EAV 从而解析有效行和无效行的标志, 由于显示系统一般为 8 bit 量化, 所以将之前配置的 12 bit 的同步字截取高 8 bit。

根据产生的有效行和无效行标志, 得到 hcnt (水平方向像素计数) 和 vcnt (垂直方向像素计数), 其中有效像素数量为 1920×1080。根据图 5 的像素阵列图, 同步字解析后可产生图像数据和行场同步信号。图像数据可直接通过移位寄存器的方式获取。

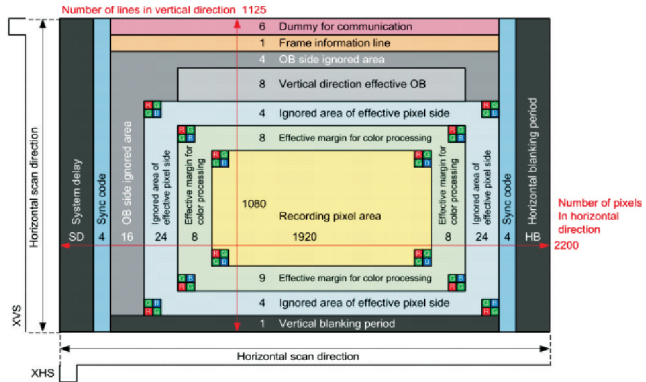


图 5 HD1080P 模式像素阵列图

行场同步信号需通过状态机的方式实现, 实现流程如表 1 所示。状态 1 为 WAIT\_INVLINE, 为等待无效行, 该状态下不需要图像数据, 直到 invalid\_flag (无效行标志) 为 1 跳转到状态 2。状态 2 为 WAIT\_VALINE, 为等待有效行的到来, valid\_flag (有效行标志) 为 1 跳转到状态 3, 状态 3 产生 vsync 也就是 1 帧图像的起始标志。valid\_flag 为 vsync 拉高的标志。当行结束同步字 EAV 到来时 vsync 拉低随即跳转到状态 4。状态 4 为 24 个无效区, 可用计数器进行检测, 当计数完 24 行图像数据时跳转到状态 5。状态 5 中启动水平方向计数器和垂直方向计数器产生有效区域标志, 根据图 5 所示水平像素的有效区域为 49 ~ 1968 一共 1920 个像素, 每检测完一行图像数据垂直方向计数器加 1 并判断当前是否完成 1080 行所有像素的检测, 如果完成检测跳转到状态 1。

## 2.2 Bayer 图像转 RGB 图像模块设计

IMX222 传感器获得彩色图像的方法是在表面覆盖 1 个只含红、绿、蓝 3 色的马赛克滤镜,对其输出信号通过一定的处理产生 Bayer 图像<sup>[4]</sup>。往外逐行输出数据时,像素的序列为 GRGRGR.../BGBGBG...。这样阵列的设计,使 RGB 传感器的色域减少全色传感器的 1/3。但是,要还原一个真正图像,需要每一个点都有 RGB 3 种颜色即 RGB 图像。因此,解析完行场同步信号后,需把像素进行 Bayer 转 RGB 的处理,由 ZYNQ 的 PL 端实现。

Bayer 转 RGB 通过 3×3 的矩阵插值计算得到<sup>[5]</sup>,由于一行图像有 1920 个 8 bit 的像素点,构建矩阵需要建立两个深度为 2048 位宽为 8 bit 的 FIFO,流程如图 6 所示。

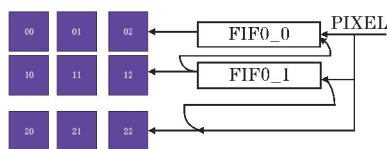


图6 双FIFO构成3×3矩阵示意图

双FIFO进行流水线操作,第1行和第2行图像数据将FIFO\_0和FIFO\_1填充完,当第3行图像数据到来时读出FIFO\_0和FIFO\_1中的数据,与此同时第3行数据写入FIFO\_1中,FIFO\_1读出的数据写入FIFO\_0中,循环往复直到最后3行数据就不再进行写入操作。

$$\left\{ \begin{array}{lll} R11 = R11 & G11 = \frac{G01 + G10 + G21 + G12}{4} & B11 = \frac{B00 + B02 + B20 + B22}{4} \quad (\text{奇行奇列}) \\ R12 = \frac{R11 + R13}{2} & G12 = G12 & B12 = \frac{B02 + B22}{2} \quad (\text{奇行偶列}) \\ R21 = \frac{R11 + R31}{2} & G21 = G21 & B21 = \frac{B20 + B22}{2} \quad (\text{偶行奇列}) \\ R22 = \frac{R11 + R13 + R31 + R33}{4} & G22 = \frac{G12 + G21 + G32 + G23}{4} & B22 = B22 \quad (\text{偶行偶列}) \end{array} \right. \quad (1)$$

## 3 视频传输存储模块设计

ZYNQ 的架构中含有 HP0 ~ 3 4 个高性能接口,用于 PL 和 PS 之间的通信和传输高速数据流,也可进行 DMA(直接内存访问)操作,通信协议遵循 AXI 协议<sup>[6]</sup>。相比一般的总线,AXI 总线具有更高的速度和更低的延迟,可高速实时的传输视频流数据。模块利用 HP 口通过 AXI 总线将 RGB 图像数据 DMA 到 DDR 中进行存储和进行数据的读写操作。AXI-HP 口对应 DDR 内存的地址映射如图 8 所示。AXI 总线有写地址通道、写数据通道、写响应通道、读地址通道和读数据通道共 5 个通道。读写通道之间相互独立,互不影响。总线按类型可

其中 3×3 的寄存器根据图像行列的奇偶顺序实现 RGB 各个分量的插值运算。插值计算过程中无法计算最外围行和列的像素点,需额外增加两行两列像素用于插值算法。因此利用 IMX222 传感器在有效图像区域外额外提供的 8 行 8 列用于图像处理特性。如图 7 所示,图中虚线框即额外插入的图像数据。

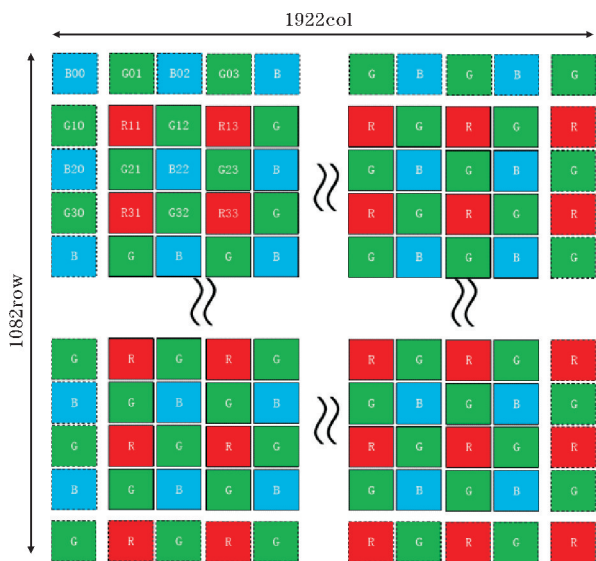


图7 Bayer 图像区结构图

图 8 中的第一个像素点 R11 已经具备 R 分量,但是 G 和 B 分量需要通过插值得到。可分为奇数行奇数列、奇数行偶数列、偶数行奇数列、偶数行偶数列 4 种情况进行插值计算,公式为

分为 AXI-Lite、AXI-Full 和 AXI-Stream 3 种。综合考虑设计复杂度和传输速度,模块采用 AXI-Full 进行设计,一次可突发 256 个 64 bit 的数据<sup>[7]</sup>。

### 3.1 AXI 总线写模块设计

将图像数据写入 DDR 内存过程中 AXI 总线写时序如图 9 所示,一次数据成功的写入首先是将数据地址传输到从机紧接着才是数据<sup>[8]</sup>。主机发送写地址有效信号(AWVALID)后,接收到从机发出的写地址准备信号(AWREADY)为 1 时,就完成一次数据地址的传输。紧接着主机发送写数据有效信号(WVALID),当接收从机发出的写准备(WREADY)信号为 1 时,就完成一次数据的传输。

Address Range	CPU's and ACP	AXI_HP	Other Bus Masters <sup>[1]</sup>	Notes
0000_0000 to 0003_FFFF <sup>[2]</sup>	OCM	OCM	OCM	Address not filtered by SCU and OCM is mapped low
	DDR	OCM	OCM	Address filtered by SCU and OCM is mapped low
	DDR			Address filtered by SCU and OCM is not mapped low
				Address not filtered by SCU and OCM is not mapped low
0004_0000 to 0007_FFFF	DDR			Address filtered by SCU
0008_0000 to 000F_FFFF	DDR	DDR	DDR	Address filtered by SCU
		DDR	DDR	Address not filtered by SCU <sup>[3]</sup>
0010_0000 to 3FFF_FFFF	DDR	DDR	DDR	Accessible to all interconnect masters
4000_0000 to 7FFF_FFFF	PL		PL	General Purpose Port #0 to the PL, M_AXI_GP0
8000_0000 to BFFF_FFFF	PL		PL	General Purpose Port #1 to the PL, M_AXI_GP1
E000_0000 to E02F_FFFF	IOP		IOP	I/O Peripheral registers, see Table 4-6
E100_0000 to E5FF_FFFF	SMC		SMC	SMC Memories, see Table 4-5
F800_0000 to F800_0BFF	SLCR		SLCR	SLCR registers, see Table 4-3
F800_1000 to F880_FFFF	PS		PS	PS System registers, see Table 4-7
F880_0000 to F8F0_2FFF	CPU			CPU Private registers, see Table 4-4
FC00_0000 to FDFE_FFFF <sup>[4]</sup>	Quad-SPI		Quad-SPI	Quad-SPI linear address for linear mode
FFFC_0000 to FFFF_FFFF <sup>[2]</sup>	OCM	OCM	OCM	OCM is mapped high
				OCM is not mapped high

图 8 地址映射图

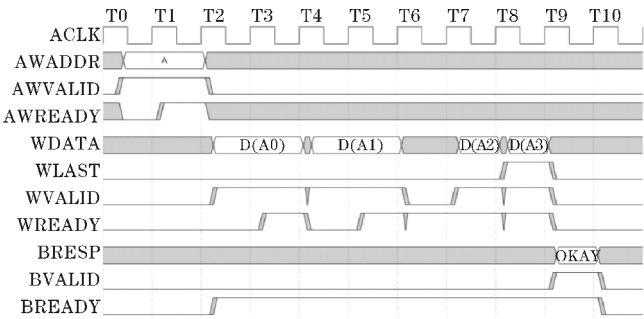


图 9 AXI 突发写时序图

由于 IMX222 传感器采集图像的时钟和 AXI 总线写入 DDR 内存中的时钟不一致,并且 RGB 图像每一个像素点的位宽为 24 bit,AXI 总线突发写数据的位宽为 64 bit。因此需插入异步 FIFO 解决跨时钟和数据位宽不匹配的问题<sup>[9]</sup>。根据 FIFO 的特性将 FIFO 写入数据位宽设置为 32 bit,有效图像数据为低 24 bit, FIFO 的读出端为 64 bit,每帧图像都有帧有效信号和行有效信号,可根据这两个信号将数据缓存 FIFO 中,当缓冲大于一次的突发数据后启动突发数据写入 DDR 中。因为每写入一帧图像需要将内存地址清零以便接受下一帧图像。每帧图像最后一次突发的起始地址为  $1920 \times 1080 \times 4 - 256 \times 8 = 8292352$ 。设计框图如图 10 所示。

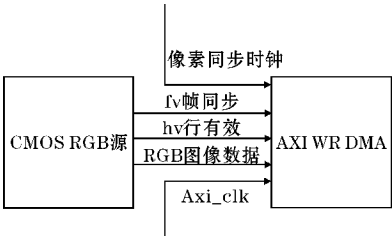


图 10 AXI 总线写模块框图

3.2 AXI 总线读模块设计

AXI 总线读模块设计与写模块类似,只是将写通道的时序变为读通道的时序。由于最终显示端是 HDMI 接口,需要根据 HDMI 的驱动时序读取视频数据。HDMI 读取数据的位宽和读取速率与 AXI 总线不一致,所以也需要插入异步 FIFO 对数据进行缓存<sup>[10]</sup>。当 FIFO 中不足一行图像书时,AXI 总线从 DDR 中读取数据进行填充,直到 FIFO 中图像数据大于一行时启动 HDMI 进行显示。设计中 AXI 总线读入 DDR 中数据的时钟大于 HDMI 显示数据的时钟,因此可以进行流畅显示,设计框图如图 11 所示。

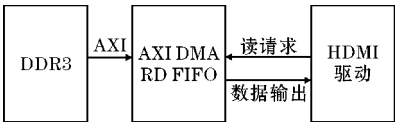


图 11 AXI 总线写模块框图

3.3 双帧缓存模块设计

在 AXI 总线对 DDR 进行读写的模块中,使用的是单个 BUFFER 缓存视频流数据。设计中 IMX222 传感器输入的帧频为每秒 30 帧,而 HDMI 显示器显示的帧频为每秒 60 帧,显示频率大于摄像头的输入频率。这样会导致显示的一帧图像上半帧为老图像,下半帧为新图像,最终产生图像的撕裂。设计通过添加双 BUFFER 进行缓存解决视频图像撕裂的问题<sup>[11]</sup>。将 2 个 BUFFER 分为 A 区和 B 区,数据源端引出一个标志信号,当图像数据正在填充 A 区时,此标志信号为 0,当图像数据正在填充 B 区时,此标志信号为 1。通过标志信号引导读出显示模块进行读取的区域,即当标志信号为 0 时读取 B 区中的数据,为 1 时读取 A 区中的数据。所以在 AXI 写模块中需一次写入两帧图像数据,两帧图像各自突发结束地址作为标志信号。这样可使用双 BUFFER 思想巧妙避开读写同一区域的问题。由于读取速度比较快,所以不会出现正在读取的这帧图像被写入数据追赶上的情况。从而显示的图像为一帧完整的图像不会出现垮帧显示,解决帧不同步问题。

4 视频显示模块设计

HDMI 系统架构由发送端和接收端两部分组成,设计使用 HDMI 发送端传输视频到显示器端。HDMI 包含 4 对差分信号线,其中 1 对传输差分时钟,另外 3 对传输音、视频数据<sup>[12]</sup>。如图 12 所示为 HDMI 的总线框图,其中 TMDS 为最小化传输差分信号<sup>[13]</sup>。

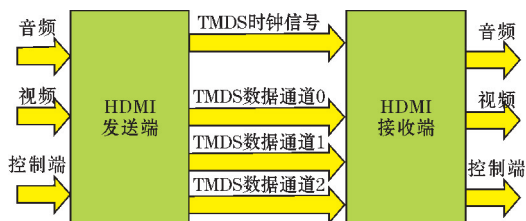


图 12 HDMI 总线框图

模块的总体框架如图 13 所示, VGA 控制器将 AXI 总线从 DDR 读取的数据转换为 R、G、B 3 个通道, 每个通道的数据位宽均为 8 bit。为保证直流平衡达到更好的显示效果, 由 8b/10b 模块将 8 bit 数据转为 10 bit, 最后由并行转串行模块将 10 bit 数据转为串行的差分信号并输出到 FPGA 的引脚端<sup>[14]</sup>。VGA 控制器色彩是由 R、G、B(红、绿、蓝)三基色组成。显示采用电子束逐行扫描的方式, 产生的 RGB 三基色合成一个彩色像素。扫描从屏幕的左上方开始, 从左到右, 从上到下, 进行扫描, 每扫完一行, 电子束回到屏幕的左边下一行的起始位置, 在这期间需要进行消隐处理。每行图像数据结束时, 用行同步信号进行同步; 扫描完所有行, 用场同步信号进行同步, 并使扫描回到屏幕左上方, 同时进行场消隐, 准备进行下一场的扫描。并转串模块使用原语 OSERDESE2 进行串并转换, 最后用原语 OBUFDS 将串行数据转换为差分信号<sup>[15]</sup>。

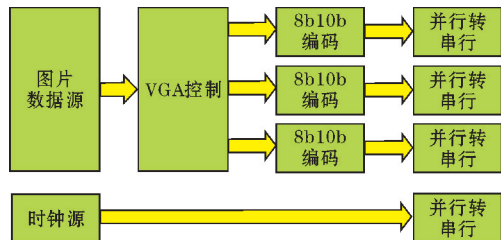


图 13 显示模块系统框架图

## 5 实验结果

该设计基于 ZYNQ 系列芯片, 在 EAGLGO 板卡上实现了对 1920×1080P 高清视频流的采集、传输、存储以及显示, 视频显示频率为每秒 60 帧且没有撕裂效果, 达到了实时高清显示的目标, 最终效果如图 14 所示。根据图 15 可知系统的功耗仅有 2.237 W, 适合移植到嵌入式设备进行分布式部署。

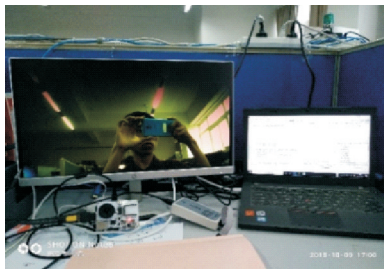


图 14 系统效果图

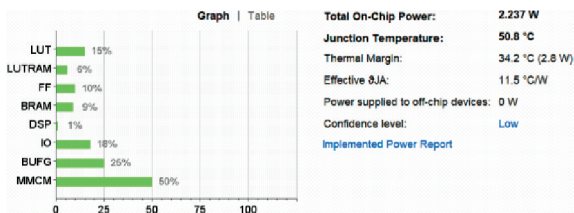


图 15 系统资源消耗以及功耗图

## 6 结束语

利用该平台可提升目前嵌入式设备视频显示的频率和分辨率, 使其达到每秒显示 60 帧 1080 P 分辨率的图像, 解决了传统嵌入式平台显示分辨率低的瓶颈。同时设计占用 PL 和 PS 的资源并不多, 因此可以根据实际需求在 PL 和 PS 部分继续添加算法和功能, 为系统进一步升级换代提供了基础。

## 参考文献:

- [1] 陈冠成, 吴贵燕. 基于 ZYNQ 芯片的实时视频处理系统设计[J]. 现代电子技术, 2019, 42(6): 76-78.
- [2] 邓兴汉, 杨恢东. 基于 FPGA 的视频图像采集系统的设计与实现[J]. 微计算机信息, 2010, 26(8): 138-139.
- [3] 贡镇. 基于 FPGA 的实时视频图像采集与显示系统的设计与实现[J]. 现代电子技术, 2013, 36(13): 46-8.
- [4] 曲金秋, 曹云峰, 许蕾, 等. FPGA 的视频流图像采集和预处理系统设计[J]. 单片机与嵌入式系统应用, 2018, 18(10): 70-74.
- [5] 张龙, 冯帅. 基于 ZYNQ 的视频图像处理平台的设计[J]. 计算机与数字工程, 2016, 44(12): 2519-2523.
- [6] 李朗, 张索非, 杨浩. 基于 Zynq-7000 的视频处理系统框架设计[J]. 计算机技术与发展, 2017, 27(5): 192-195.
- [7] 宁效龙, 何子力, 张昕昱, 等. 基于 Zynq 与 Qt 的视频采集与图像边缘检测系统[J]. 信息技术与网络安全, 2019, 38(2): 71-74.
- [8] 孟繁星. 基于 ZYNQ-7000 的视频图像处理系统设计[J]. 信息通信, 2015(8): 60-62.
- [9] 王浩宇, 漆晶, 谭歆. 基于 Zynq-7000 的视频采集平台设计[J]. 工业控制计算机, 2017, 30(4): 55-56.

- [10] 阮远忠,喻金华,肖铁军. 基于 ZYNQ-7000 的视频图像处理系统设计与实现[J]. 软件导刊, 2018,17(9):148-152.
- [11] 唐斌. 基于 FPGA 的双端口视频缓冲器设计[J]. 液晶与显示,2014,29(4):575-579.
- [12] 黄国鹏,刘卫东,乔明胜,等. 基于 FPGA 的 LVDS 视频图像采集与预处理系统的设计实现[J]. 现代显示,2009(2):32-35.
- [13] 褚亭强. 基于 ZYNQ 的 HDMI 显示端设计与实现[J]. 电视技术,2017,41(2):23-27.
- [14] 潘磊,葛中芹,庄建军,等. 基于 FPGA 的 HDMI 视频流图像处理的系统设计[J]. 实验室研究与探索,2015,34(10):76-80.
- [15] 张龙,冯帅. 基于 ZYNQ 的视频图像处理平台的设计[J]. 计算机与数字工程,2016,44(12):2519-2523.

## Frame Design of Video Processing Platform based on ZYNQ

WAN Yao<sup>1</sup>, ZHOU Yong<sup>2</sup>

(1. College of Communication Engineering, Chengdu University of Information Technology, Chengdu 610225, China; 2. College of Communication Engineering, Chengdu Technological University, Chengdu 611730, China)

**Abstract:** With the development of machine vision, a real-time video acquisition system based on Xilinx ZYNQ chip is proposed for the problems of slow video data acquisition speed, low resolution and low real-time performance of current embedded devices. The heterogeneous design of FPGA+ARM uses this feature to build a video processing platform designed by hardware and software. The FPGA side realizes the image preprocessing, the ARM side realizes the configuration of the peripherals, and the image transmission uses the AXI bus to store the data in the DDR at high speed through the HP port, and finally uses the HDMI port to display the video. By using the parallelism of FPGA and the high-performance bus in the chip, the system has lower power consumption and faster processing speed, larger bandwidth, and real-time performance compared to general embedded devices. It provides a reliable and stable platform for distributed deployment of monitoring systems and implementation of more complex image processing algorithms.

**Keywords:** communication integrated circuit; embedded system; SPI; IMX222; ZYNQ; AXI bus; FIFO