

一种可编程异构芯片设计方法应用于视频桥接

王潘丰^{1,2}, 蔡懿慈¹

(1. 清华大学计算机科学与技术系, 北京 100084; 2. 京微齐力(北京)科技股份有限公司, 北京 100190)

摘要: 随着智能时代的到来,越来越多的设备拥有摄像头和显示屏,而它们具有各种各样不同接口和视频格式,视频桥接面临新的挑战。以往的解决方案是根据接口和视频格式的需求采用不同的电路,如现场可编程门阵列(Field Programmable Gate Array, FPGA)、图形处理器(Graphics Processing Unit, GPU)和专用集成电路(Application Specific Integrated Circuit, ASIC)等。但这种单一的电路模式难以同时满足低成本、超低功耗和小型化的要求,尤其是在移动显示领域。本文提出了一种新的异构体系架构,它将FPGA、微控制单元(MicroController Unit, MCU)、ASIC和存储器无缝集成到一个芯片中。该芯片不仅实现了小型化,而且具有低成本和低功耗的优势;更重要的是该款芯片可以支持不同接口和视频格式的桥接需求。针对不同算法的应用,本文给出了使用该芯片的评估方法和解决方案,为架构设计提供了依据。该芯片已成功在22 nm工艺流片,整体尺寸约为4 mm×4 mm,总功耗约为200 mW。它可以支持3 840×2 160分辨率和144 Hz刷新率的视频输入格式,1 080×2 340分辨率和90 Hz刷新率的视频输出格式。在实现同样视频桥接功能的应用时,本文所提芯片的面积和功耗均小于AMD芯片XC7K325T和Zynq Z7035的1/10。换言之,针对此类场景的应用,本文方案在成本和功耗方面相比于传统商业FPGA有显著优化。

关键词: 异构架构;可编程;现场可编程门阵列(FPGA);专用集成电路(ASIC);视频桥接;低功耗

基金项目: 国家自然科学基金(No.62220106011)

中图分类号: TN402

文献标识码: A

文章编号: 0372-2112(2025)01-0072-12

电子学报URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20240144

A Programmable Heterogeneous Chip Designed for Video Bridging

WANG Pan-feng^{1,2}, CAI Yi-ci¹

(1. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China;

2. Hercules Microelectronics, Co., Ltd., Beijing 100190, China)

Abstract: With the development of intelligent era, more and more devices have cameras and display screens, which are in various video formats with different interfaces. To fill the gaps, video bridging is widely required. The previous solutions adopted field programmable gate array (FPGA), graphics processing unit (GPU), and application specific integrated circuits (ASIC). However, it is difficult to meet the requirements of low cost and ultra-low power consumption and miniaturization, especially in the field of mobile display. This paper proposes a novel heterogeneous architecture which seamlessly integrates FPGA, microcontrol unit (MCU), ASIC, and memory into a single silicon chip. This chip not only achieves miniaturization, but also has the advantages of low cost and low power consumption; More importantly, this chip can support bridging requirements for different interfaces and video formats. At the same time, this paper provides evaluation methods and solutions for different algorithm applications, and provides a basis for architecture design. The chip has been successfully taped out in an industrial 22 nm process. It can support video input formats with a resolution of 3 840×2 160 and a refresh rate of 144 Hz, as well as video output formats with a resolution of 1 080×2 340 and a refresh rate of 90 Hz. The experimental results show that, in supporting the similar function, the overall chip size is about 4 mm×4 mm and the total power consumption is about only 200 mW, both of which are less than one tenth of AMD XC7K325T and Zynq Z7035. In other words, for applications in video bridging scenarios, our solution has significant optimization compared to traditional commercial FPGAs in terms of cost and power consumption.

Key words: heterogeneous architecture; programmable; Field Programmable Gate Array (FPGA); Application Specific Integrated Circuit (ASIC); video bridging; low power

Foundation Item(s): National Natural Science Foundation of China (No.62220106011)

1 引言

随着高速通信技术的发展和人工智能技术的不断进化,智能时代已经到来.手机、汽车、手表、家电、无人机、安防监控等,都在向智能化迈进,最重要的表现之一是显示屏和摄像头的数量不断增加.嵌入式移动处理器的功能越来越强大,图像传感器和显示器的成本也越来越低.在终端产品的开发中,每个传感器和显示屏都希望连接到处理器,但2个接口之间往往存在不匹配的场景,如格式不同、分辨率不同、带宽不足等.因此,视频桥接芯片应运而生,并被广泛使用.

视频桥接功能有多种实现方案,如现场可编程门阵列(Field Programmable Gate Array, FPGA)、图形处理器(Graphics Processing Unit, GPU)、中央处理器(Central Processing Unit, CPU)、专用集成电路(Application Specific Integrated Circuit, ASIC)、多处理器片上系统(MultiProcessor System on Chip, MPSoC)等.赛灵思(Xilinx)的XC7K325T芯片用于实现从通用串行总线(Universal Serial Bus, USB)摄像头到外围组件互连高速(Peripheral Component Interconnect express, PCIe)接口的视频数据传输^[1].在热成像视频应用中,多个FPGA芯片用于实现高清晰度串行数字接口(High Definition - Serial Digital Interface, HD-SDI)到PCIe的传输^[2].Microsemi公司的FPGA芯片SmartFusion2,用于物联网(Internet of Things, IoT)领域的视频处理和传输^[3].Xilinx FPGA Artix-7用于实现相机图像的采集和处理,然后从PCIe接口输出^[4].用Xilinx Zynq FPGA来实现视频桥接过程^[5-8].采用CPU和GPU来实现图像处理和传输^[9-11],以及将Xilinx FPGA VU35P与IBM公司的CPU芯片Power9相结合,通过PCIe进行视频处理和传输^[12].

综上的视频桥接方案多种多样,但都存在一定的不足.如CPU和GPU具有最高的灵活性,但并行处理能力不足,外部传输接口较差,二者的结合需要数据共享,这既昂贵又低效;FPGA具有最好的并行性,但成本高,资源利用率低;CPU+FPGA架构合理,但通常需要许多片外存储器访问,功耗较大,效率不高^[13];Xilinx MPSoC将CPU和FPGA集成在单个芯片中,提高了效率,但它对用户的开发并不友好,成本太高,并且它需要配备高功耗的外部双倍速率同步动态随机存储器(Double Data Rate, DDR)^[14].

在工业市场上,有一些ASIC芯片可以实现视频桥接功能.但由于需求的多样性,往往需要一系列芯片,这增加了板级开发和产品线管理的成本.嵌入式FPGA是另一种解决方案.2002年,IBM和Xilinx的工程师提

出了ASIC和FPGA的混合架构^[15],它可以同时利用二者的优势.

由此可见,视频桥接领域需要新的技术方法,其中可编程的异构架构是一个方向,FPGA和ASIC不断相互渗透是一种趋势.FPGA引入了包括处理器在内的多种模块,而传统的SoC开始集成嵌入式FPGA模块.对于一些特定领域的应用可以对架构进行优化,取得了很好的效果^[16-21].在视频桥接领域,本文提出了一种评估方法来寻找更合适的解决方案.本文的主要贡献如下:

(1)提出了一种新的用于视频桥接的可编程异构架构,它将FPGA、微控制单元(MicroController Unit, MCU)、ASIC和存储器集成到一个芯片中.该芯片可以支持多种不同视频桥接需求.

(2)提出了一种针对视频桥接算法的评估方法,基于该方法给出了应用该芯片解决视频桥接问题的实现方案,提供了早期架构设计评估的依据.

(3)该芯片经过完整的设计和验证,并采用22 nm工艺成功流片并量产.该芯片的面积仅约4 mm × 4 mm,不到XC7K325T的1/10;功耗仅为200 mW左右,小于XC7K325T及Z7035的1/10.优化结果显著.

2 视频桥接常用算法介绍及分析

视频桥接中的算法通常由源和目的地的视频格式决定.有时为了节省内存,需要进行压缩和解压缩.视频桥接应用中的一些典型数据流如图1所示.其中,RX是接收模块,TX是发送模块,ENC是压缩模块,DEC是解压缩模块,CSC是色彩空间转换模块,Scaling是缩放模块,SRAM是存储模块.图1中的数据流(a)~(e)分别对应于不同的应用场景.从图中可以看出,除去接收和发送的接口模块以及必要的存储模块,视频桥接领域常用的算法有3种:编解码算法、色彩空间转换算法和缩放算法.本节将对上述3种算法的原理进行简单的分析和介绍.

2.1 编解码算法

编解码算法有很多种,在本文中,以显示流压缩(Display Stream Compression, DSC)标准为例进行分析.DSC标准是一种为高分辨率、有限带宽应用定义的,实时视觉无损压缩算法的标准^[22],其基本思想是差分脉冲编码调制(Differential Pulse Code Modulation, DPCM)^[23].根据视频电子标准协会(Video Electronics Standards Association, VESA)DSC标准^[24],编码和解码的主要架构分别如图2和图3所示^[25].DSC标准解压缩算法包括像素的预测量化和重构.

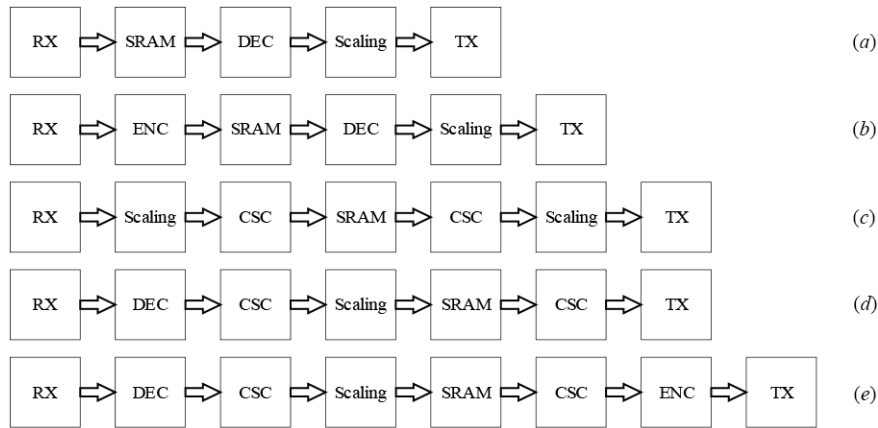


图1 视频桥接应用典型数据流

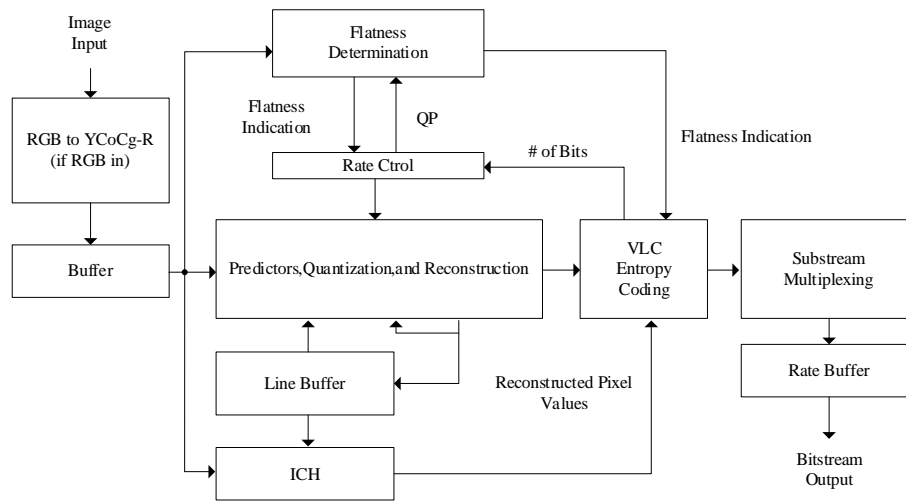


图2 DSC 编码架构图

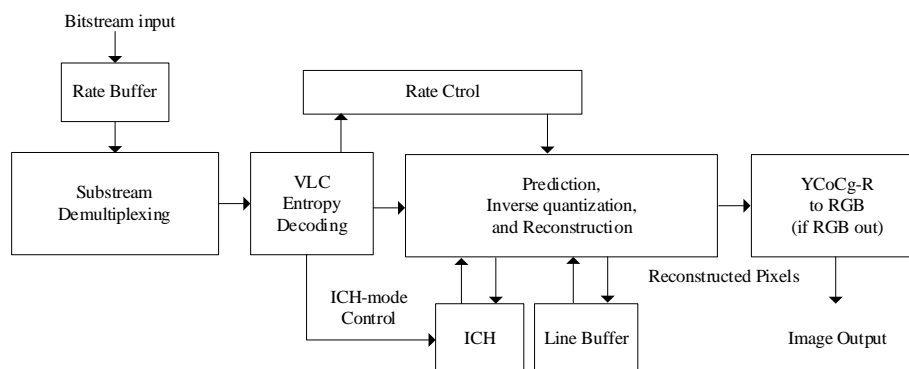


图3 DSC 解码架构图

2.2 色彩空间转换算法

在色彩空间中有2种常见格式——RGB和YUV。在RGB色彩空间中,红色、绿色和蓝色是基本元素。RGB格式是图像采集和显示中常用的格式。在YUV色彩空间中,颜色分为3个分量:“Y”表示亮度,也称为灰度值;“U”和“V”表示色度,用于描述图像的颜色和饱和

和度。

YUV还有另一种形式——YCbCr。其中,“Y”除了与YUV中包含亮度信息的含义相同,还包含G(绿色)分量信号。“Cb”和“Cr”也指颜色,但它们在表达上不同。“C”表示分量,“Cb”和“Cr”分别对应B(蓝色)和R(红色)分量信号。YUV格式有3种主要的流采样方法,

YUV4:4:4、YUV4:2:2和YUV4:2:0. 对于YUV4:4:4, 每个Y对应一组UV分量;对于YUV4:2:2,每2个Y共享一组UV分量;对于YUV4:2:0,每4个Y共享一组UV分量. RGB色彩空间通常用于相机和监视器,而YCbCr颜色空间通常被用于视频压缩和传输等处理^[26,27].

基于ITU-R BT601协议^[28],YCbCr到RGB的转换公式如式(1);反过来,RGB到YCbCr的转换公式如式(2).

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0.000 & 1.596 \\ 1.164 & -0.392 & -0.813 \\ 1.164 & 2.017 & 0.000 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & -0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (2)$$

2.3 图像缩放算法

图像缩放通常通过线性插值算法来实现. 常用的线性插值算法有最近邻插值、双线性插值和双三次插值^[29-31].

2.3.1 最近邻插值

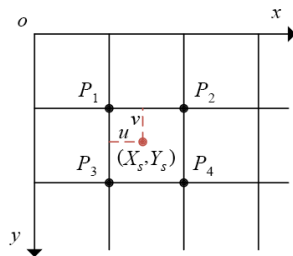
最近邻插值的基本思想是直接复制最近像素的值作为要插入的像素值. 坐标计算如下:

$$\begin{cases} X_s = X_d \times \frac{W_s}{W_d} \\ Y_s = Y_d \times \frac{H_s}{H_d} \end{cases} \quad (3)$$

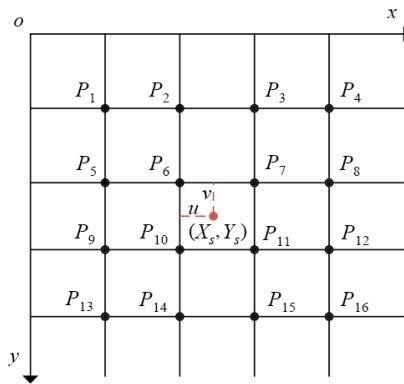
其中, X_d, Y_d 是目的像素的坐标, X_s, Y_s 是源图像中的坐标, W_s, H_s 是源图像的宽度和高度, W_d, H_d 是缩放图像的宽度和高度.

2.3.2 双线性插值

双线性插值算法的基本思想是使用待插值像素周围4个像素值的加权和作为目标像素值,计算过程如下.



(a) 双线性插值



(b) 双三次插值

图4 双线性插值和双三次插值

首先,计算源图像中像素的映射坐标. 取源图像映射点周围的4个点,如 P_1, P_2, P_3 和 P_4 ,如图4(a)所示. 在图4中, u 和 v 分别是在水平方向和垂直方向上从映射点到像素点 P_1 的距离.

其次,计算权重,它与映射点到2个方向上每个点的距离有关. 权重计算如下:

$$w_{P_1} = (1-u)(1-v) \quad (4)$$

$$w_{P_2} = u(1-v) \quad (5)$$

$$w_{P_3} = (1-u)v \quad (6)$$

$$w_{P_4} = uv \quad (7)$$

因此,可以计算像素值 $P(X_d, Y_d)$:

$$P(X_d, Y_d) = w_{P_1}P_1 + w_{P_2}P_2 + w_{P_3}P_3 + w_{P_4}P_4 \quad (8)$$

2.3.3 双三次插值

双三次插值算法的思想类似于双线性插值. 不同之处在于双三次插值考虑了图4(b)所示的映射点周围的16个像素,并且通过式(9)中的插值函数 $S(x)$ 和式(10)中的像素值 $P(X_d, Y_d)$ 来计算权重.

$$S(x) = \begin{cases} |x|^3 - 2|x|^2 + 1, & |x| \leq 1 \\ -|x|^3 + 5|x|^2 - 8|x| + 4, & 1 < |x| \leq 2 \\ 0, & \text{其他} \end{cases} \quad (9)$$

$$P(X_d, Y_d) = \begin{bmatrix} S(1+u) \\ S(u) \\ S(1-u) \\ S(2-u) \end{bmatrix}^T \begin{bmatrix} P_1 & \cdots & P_4 \\ \vdots & \ddots & \vdots \\ P_{13} & \cdots & P_{16} \end{bmatrix} \begin{bmatrix} S(1+v) \\ S(v) \\ S(1-v) \\ S(2-v) \end{bmatrix} \quad (10)$$

从以上分析可以看出:最近邻插值算法计算量最小、速度最快,但缩放后的马赛克现象严重^[32],缩放效果最差;双三次插值算法具有最好的缩放效果,但计算量最大;双线性插值算法的效果和计算量相对折衷. 应根据具体应用的不同需求选择适当的算法. 例如,如果缩放效果是可接受的,则最近邻插值算法最合适;而如

果对缩放效果有很高的要求,并有足够的计算资源,则双三次插值算法是最适合的算法。

2.4 本文研究内容

本文研究的问题是如何灵活地支持上述不同算法的应用,如何为视频桥接领域的应用找到更好的解决方案,以获得统一化、小型化以及更低的成本和功耗。为了实现图1中的所有视频桥接应用,芯片的功能框图如图5所示。图1中的所有功能模块都包括在内,此外增加了控制器 Controller 负责数据流的管理。

FPGA 比 ASIC 更灵活,而 ASIC 与 FPGA 相比具有更低的成本和功耗,混合架构正变得越来越流行。设计的挑战在于如何分析算法并对其进行分解,如何选择合适的实现方法,如何将各种资源集成到单个芯片中并确保一定的灵活性。

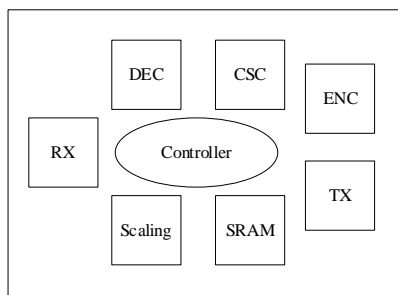


图5 功能模块框图

3 可编程异构芯片架构

本文提出了一种可编程异构芯片架构,它将 FPGA、MCU、ASIC 和存储器集成到一个芯片中,可以支持不同接口和视频格式的桥接需求。本节将讨论整体芯片架构、各个组件以及它们之间的互连。

3.1 体系架构概述

可编程异构体系架构如图6所示,其技术创新性在于将 FPGA、MCU、ASIC 和存储器集成在一颗单一芯片中。该体系架构结合了它们各自的优势,通过平衡性能、功耗和面积来实现全局最优解决方案。当前市场上与该体系架构最相似的芯片是 Xilinx 的 Zynq 系列芯片,其性能及可编程性能突出,但缺乏固化的核心算法模块 ASIC,因此在功耗和面积成本上没有优势。本文提出的架构既有 FPGA 又有 ASIC,根据模块不同的属性而采用最适合的实现形式。图6中以不同颜色标注,黄色是 FPGA 部分,其余为 ASIC 部分。

数据流的控制是在 FPGA 和 MCU 中实现的。FPGA 负责数据路径的设置,每个模块都连接到 FPGA, FPGA 本身可以进行硬件编程。因此,通过对 FPGA 进行编程,可以根据图1中的不同应用场景来调整数据路径。数据流的控制部分由 MCU 进行管理。例如,每个模块的初始化、模块之间的握手、空/满检测、异常处理等,都

可以通过 MCU 中的软件编程轻松实现。因为 FPGA 的嵌入,该体系架构比纯 ASIC 具有更强的灵活性。

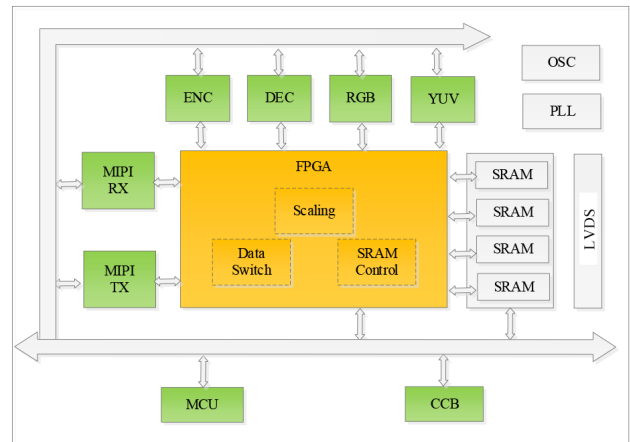


图6 可编程异构架构

3.1.1 FPGA

FPGA 是该体系架构的核心,用于实现图像缩放算法、数据切换、SRAM 控制和其他粘合逻辑。FPGA 架构框图如图7所示。其中,PLB×10 表示10个可编程逻辑块(Programmable Logic Block, PLB),DSP×2 表示2个数字信号处理器(Digital Signal Processor, DSP), EMB×2、VMB×2 同理。

PLB 包括8个6输入查找表(Look-Up-Table, LUT)和寄存器。PRB 是 PLB 加上本地 SRAM。EMB 是由4个9 kbits SRAM 组成的嵌入式存储块。它支持不同的位宽,如 x1、x2、x4、x9、x18,以及宽度和深度的扩展。此外,它还具有纠错码(Error Correction Code, ECC)和先进先出(First In, First Out, FIFO)等其他功能。VMB 是视频存储块,它是在传统 EMB 的基础上为视频桥接应用而优化的。VMB 有2个4 kBytes SRAM,仅支持 x32 和 x64 的位宽。DSP 是一个嵌入式乘法器和累加器,支持1个 35×18、2个 18×18 或4个 9×9 乘法运算。

为了在 FPGA 中实现第2节中提到的视频缩放算法、数据流控制和 SRAM 控制器功能,首先构建一个虚拟器件,将这些功能用硬件描述语言实现;再通过计算机辅助设计软件(EDA 工具)进行综合,并在此器件上进行布局布线。根据结果所显示的资源使用情况,再调整虚拟器件中的各类资源数量。经过迭代,最终确定 FPGA 部分所包含的资源为5K个6输入 LUT、8×36 kbits EMB、16×8 kBytes VMB 和8个 DSP。

3.1.2 硬核 ASIC

除可编程电路外,此架构集成了一些硬核加速器,用来实现第2节中提到的一些算法,如 ENC、DEC 和色彩空间转换模块(RGB 和 YUV)。此外,MIPI TX 和 RX、低电压差分信号(Low Voltage Differential Signaling,

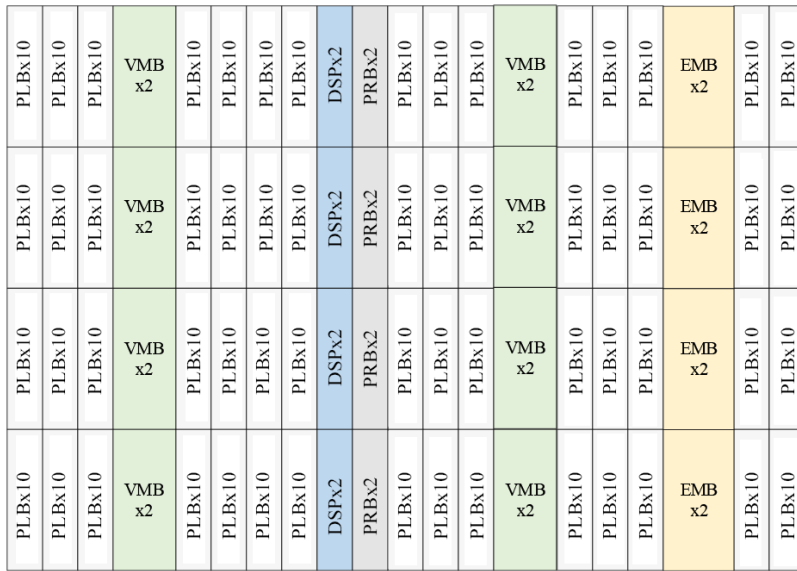


图7 FPGA架构

LVDS)接口和显示串行接口(Display Serial Interface, DSI)等外部接口都在ASIC中实现.

3.1.3 MCU

MCU包括嵌入式处理器和一些常见的外围设备,如通用异步收发器(Universal Asynchronous Receiver/Transmitter, UART)、通用输入/输出(General Purpose Input Output, GPIO)、I2C、串行外设接口(Serial Peripheral Interface, SPI)、直接内存访问(Direct Memory Access, DMA)、定时器、看门狗定时器(Watch Dog Timer, WDT)等. MCU负责控制内部模块和与外部处理器的通信,这对于用户开发他们的应用系统是友好和方便的.

3.1.4 片上存储器

该存储器用作视频帧的数据缓存. 以往的许多解决方案都使用外部存储器,使得数据访问速度慢、功耗高. 此架构中采用内部嵌入存储器可以加快存储器访问速度并扩展带宽. 随着智能网络和显示技术的发展,高清(High Definition, HD)、全高清(Full High Definition, FHD)和超高清(Ultra High Definition, UHD)分辨率越来越受欢迎. 在本文中,我们的目标是支持HD和FHD分辨率. FHD图片的常见大小约为2.5 MB,因此片上集成了5 MB的SRAM,这样可以提供2帧的存储容量,便于乒乓操作.

此外,它还包括片上晶体振荡器(OSC)、锁相环(PLL)、配置控制块(CCB)和其他基本模块. 此处不作赘述.

3.2 互连结构

异构体系架构中有许多功能模块,如何将它们连接起来是集成的关键. 下面分2个部分进行介绍,一部

分是FPGA与其他模块的连接,另一部分是片上总线矩阵连接.

3.2.1 FPGA连接

为了支持图1中的不同数据流,所有算法块都需要与FPGA连接,FPGA是数据传输的交互中心. FPGA中有不同跨越长度(1/2/3/4/8)的连接线. 通常情况下,它们在边界处会环回回去,保证连线的一致性. 长度为3的环回线如图8所示. 其中,“te”表示向东的连线(跨度为3),“tw”表示向西的连线. 切断边界处原始的环回线,并将其作为FPGA的输入和输出引脚,用于连接ASIC模块的信号,如数据、握手、中断等.

3.2.2 片上总线矩阵

除了FPGA的直接连接外,MCU、ASIC、存储器和FPGA之间的连接由ARM AMBA总线矩阵实现. 该总线矩阵有2个层次结构,AHB和APB. AHB总线上有多个主控制器(如DMA、FPGA主控制器0、FPGA主控制器1等),以及多个从控制器(如CCB、SRAM 0、SRAM 1、CSI、DSI、DMA、AHB2APB等). 在APB总线上有一些低速外设,如GPIO、SPI、WDT、I2C等. 这个总线矩阵负责参数设置和数据共享,以及用于数据传输、内存初始化、寄存器访问等.

3.3 低功耗设计

为了降低功耗,采用了一些低功耗设计,包括时钟门控、睡眠模式、电源开关等. 每个时钟域都有自己的时钟门控,用来减少动态功耗. 当MCU空闲时,它可以进入睡眠模式以节省功耗. 片上所有SRAM都可以设置为睡眠模式. 当处于睡眠模式时,SRAM中的内容被保持并且功耗很低. SRAM的电源也可以关闭,在内容不需要保持的情况下节省更多的电力.

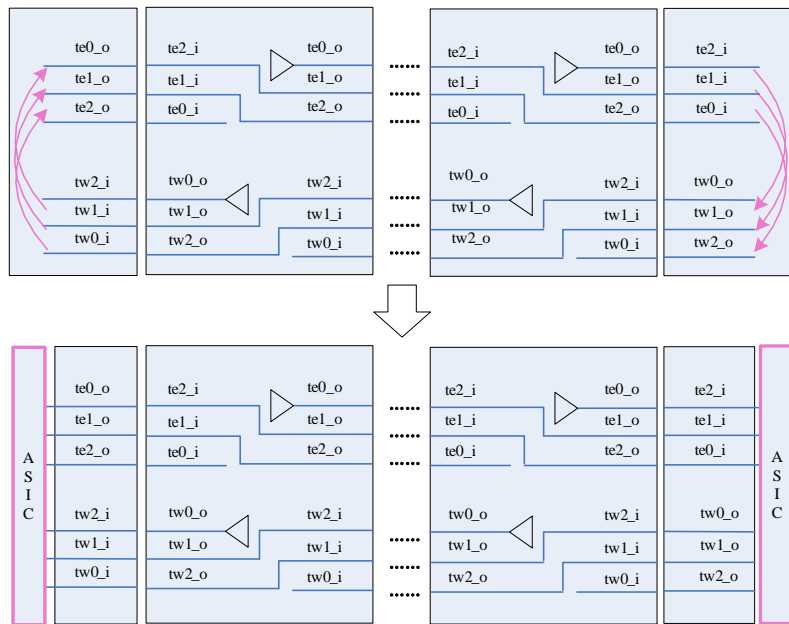


图8 FPGA与ASIC连接

对于 ASIC 模块,进行选择性配备电源开关.在实际应用场景中,一些模块需要始终打开,因此不必添加电源开关,如 MIPI RX/TX 接口;有些模块太小,不值得插入电源开关,如 CSC. 在本文的芯片中,只有 DEC 和 ENC 插入了电源开关. 这 2 个模块可以根据应用要求开启或关闭电源.

3.4 小型化设计

首先,基于该体系架构,大多数模块在 ASIC 中经过硬化处理,只嵌入了一小部分 FPGA 模块,芯片集成度很高,为小型化提供了前提基础.

其次,通过采用晶圆级芯片级封装(Wafer Level Chip Scale Packaging, WLCSP)技术,最终的芯片尺寸与硅片面积尺寸一样小,并且厚度很薄. 这种芯片可以嵌入一些狭窄的空间中,满足小型化的要求,非常适合应用于移动显示终端设备,因为在这些场景中追求轻便、小巧.

4 评估方法及解决方案

在应用本文的芯片时,如何选择片上的不同资源实现第 2 节所述算法,有不同的设计方案方法,如 CPU 和 GPU 可编程性更好, FPGA 和 ASIC 实时性更好. 在视频桥接应用领域,实时性通常非常重要和必要. 结合本文提出的可编程异构芯片设计,重点讨论 FPGA 和 ASIC 的评估和实现方法.

按照如图 9 所示流程,对算法进行分析和分类. 首先,总结了视频桥接领域中常用的算法. 其次,提取并分解成细粒度的小算法单元. 在前面分析的基础上,得到了 3 个算法单元,即压缩解压缩算法 DSC、色彩空间转换算法 CSC 和缩放算法. 最后,评估它是在 FPGA 中

实现还是在 ASIC 中实现. 通常情况下,这个决定是根据以往的经验主观作出的.

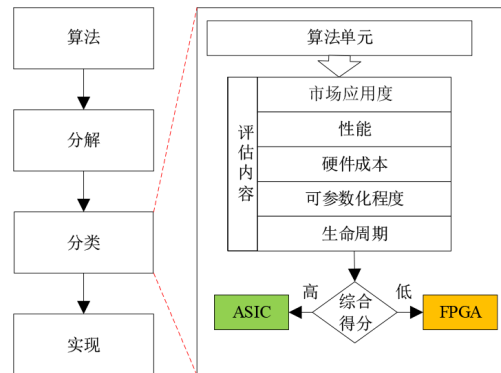


图9 算法实现评估流程

本文采用评分模型的综合评估方法进行评估. 这种评估方法的基本思想是将不同维度的评估结果综合起来,通过加权系数来反映各个维度的重要性,最后通过求和得到总的评价结果. 此方法可以给出相对更加准确和客观的评估结果,其挑战在于应该考虑哪些因素以及如何量化这些因素. 具体步骤如下.

首先,确定评估维度. 分别从 5 个维度进行评估,包括市场应用度、性能、硬件成本、可参数化程度和生命周期等.

其次,根据各个维度的重要性分配加权系数. 加权系数可以根据专家意见、历史数据或业务需求确定. 这些系数应该满足总和为 1 的条件,以确保评价结果的合理性.

然后,对每个维度进行单独评估,并得到此维度下的评分. 评估可以是定量的,也可以是定性的,但最终

需要转换为一个可以量化的分数. 为了简化模型, 每个项目都采用百分制进行评估.

最后, 加权求和, 将每个维度的得分乘以其对应的加权系数, 得到加权得分, 将所有维度的加权得分相加, 得到最终的综合评价结果.

4.1 市场应用度

如果一个算法可以在不做任何修改的情况下应用于许多场景, 即可以广泛应用于市场中, 那么它的市场应用度就很高. 对于具有高市场应用度的算法, 如无线通信标准、加密算法等, 这些算法一旦成为行业标准, 即整个行业必须共同遵守, 用户数量巨大. 一个广泛使用的例子是 SHA-256 加密算法, 该算法在区块链挖矿中非常普遍. 为这种算法设计的 ASIC 可以提供比通用 CPU 或 FPGA 更高的哈希率, 更低的功耗和成本. 对于市场应用度较低的算法, FPGA 更具优势, 因为 FPGA 可以在不影响硬件的情况下重新配置, 适应不同的算法和标准. 例如在机器学习领域, 卷积神经网络算法经常更新和改进^[33]. 使用 FPGA, 研究人员可以快速实现新的算法, 调整参数, 并在硬件上进行实验, 而不需要等待定制 ASIC 的制造.

本文采用了实地考察的调研方法. 通过实地走访, 与市场销售业务人员沟通, 以及与研发人员座谈等方式, 收集产品需求信息和市场反馈数据, 并将这些实际市场应用数据进行整理、提取和综合分析, 作为评估的基本参考数据和依据. 我们调研了 8 个商业客户, 这些商业客户的产品销往全球, 并占据重要市场份额, 因此具有一定的代表性. 调查结果表明, CSC 有 8 个用户使用. 许多用户也使用视频缩放, 但调查显示, 它们的大小和形状不同. 使用完全相同算法的用户数量很少. 有 2 个用户正在使用 DSC, 还有其他 2 个用户计划使用 DSC. 根据所有的结果, 设置最高分 100 意味着 8 个用户, 那么可以得到 DSC 对应 50, CSC 对应 100, 缩放对应 12, 如表 1 所示.

表 1 市场应用度评估分值

| 算法 | DSC | CSC | 缩放 |
|--------|-----|-----|----|
| 用户数量/个 | 4 | 8 | 1 |
| 分值 | 50 | 100 | 12 |

需特别说明的是, 这里采用的用户数量的计算方法不够完全准确, 因为这里的用户数量是公司数, 而每家公司又有他们自己的用户, 他们的用户数量是各不相同的, 理想状态下应该是计算出最终用户数, 这样才更具有意义. 然而, 此部分数据涉及商业机密, 暂时无法公开, 因此本文中默认各公司的用户数量基本相当, 只做粗略估算.

4.2 性能

与 ASIC 相比, FPGA 在可编程性方面取得了优势,

但同时以降低性能为代价^[34]. 如果性能要求很高, 而 FPGA 很难满足, 那么使用 ASIC 更合适. 性能主要取决于工作频率和吞吐率的需求.

本文的目标是支持实时 FHD 视频桥接应用. 以 $1\ 080 \times 2\ 400$ 的分辨率为例, 刷新率为 120 Hz, RGB 数据宽度为 8 位, DSC 数据宽度为 32 位. 在数据流中存在诸如行同步和帧同步之类的额外开销, 保留 15% 的余量. 然后我们可以计算 DSC 的工作频率为 $1\ 080 \times 2\ 400 \times 120 \times 8 \times (3/32)/0.85=274.447$, 即 DSC 需要在 275 MHz 下工作. CSC 需要再高出 10% 才能保证数据流不被阻塞, 约为 300 MHz. 因为数据被缓冲在 SRAM 中, 所以缩放运算的频率不必非常高, 150 MHz 就足够了. 此处以 300 MHz 作为最高频率进行打分, 可以得到 CSC 的得分为 100, 而 DSC 为 90, 缩放为 50, 如表 2 所示.

表 2 性能评估分值

| 算法 | DSC | CSC | 缩放 |
|--------|-----|-----|-----|
| 频率/MHz | 275 | 300 | 150 |
| 分值 | 90 | 100 | 50 |

4.3 硬件成本

硬件成本是指实现算法所需的芯片面积. 对于仅包含基于 LUT 的逻辑和触发器的电路, 在 FPGA 和 ASIC 中实现它们所需的硅片面积比平均为 35^[35]. 为了实现相同的功能, ASIC 可以比 FPGA 节省更多的面积. 换句话说, 当算法非常复杂时, 硬件成本较大, 更应该选择 ASIC; 否则, FPGA 实现的成本将是不可接受的.

在使用 22 nm 工艺库进行逻辑综合后, 得到了算法的面积报告. DSC 的芯片面积为 $746\ 276\ \mu\text{m}^2$. CSC 的为 $7\ 709\ \mu\text{m}^2$. 缩放(双线性插值算法, 输入最大分辨率为 $1\ 080 \times 2\ 400$, 输出最大分辨率为 $1\ 080 \times 2\ 340$)的为 $15\ 387\ \mu\text{m}^2$. 此处, 以最大面积 $746\ 276\ \mu\text{m}^2$ 为满分 100, 可以得到 DSC 得分为 100, 而 CSC 为 1, 缩放为 2, 如表 3 所示.

表 3 硬件成本评估分值

| 算法 | DSC | CSC | 缩放 |
|---------------------|---------|-------|--------|
| 面积/ μm^2 | 746 276 | 7 709 | 15 387 |
| 分值 | 100 | 1 | 2 |

4.4 可参数化程度

设计中的可参数化程度代表了算法的灵活性. 它是指寄存器传输级别(Register-Transfer Level, RTL)设计的参数, 参数越多就越灵活, 硬件电路本身已经可以适应不同的参数输入, 那么就适合采用 ASIC. 因为设计本身具有一定的灵活性, 可以取代 FPGA 的部分灵活性.

DSC 有 25 个参数, 包括图片参数设置、每行切片数、输出模式格式等; CSC 有输入格式和输出格式 2 个参数; 缩放算法具有垂直和水平 2 个参数. 此处以 25 个

参数作为最高级别的 100 分,可以得到 DSC 为 100, CSC 为 8,缩放为 8,如表 4 所示.

表 4 可参数化评估分值

| 算法 | DSC | CSC | 缩放 |
|--------|-----|-----|----|
| 参数个数/个 | 25 | 2 | 2 |
| 分值 | 100 | 8 | 8 |

4.5 生命周期

算法的生命周期反映了它的更新频率. 如果算法不固定,而且变化频繁,那么使用 FPGA 更合适;相反,ASIC 更合适. 因此,生命周期越长,就越适合使用 ASIC.

DSC 的 1.1 版于 2014 年标准化,1.2 版于 2016 年修订. 到目前为止,它被广泛使用. 用户调查显示,在不久的将来,它可能会被越来越多地使用. 估计生命周期在 10 年以上,这里假设为 10 年. CSC 算法自 1995 年以来一直使用到现在. 它主要基于 RGB 和 YCbCr 的定义,生命周期应该持续很长时间,假设为 50 年. 关于缩放算法,随着越来越多的显示器采用不同的形状和分辨率,尤其是在移动消费市场,它往往需要改进和调整,其生命周期约为 3 年. 此处将 50 年作为最长的生命周期,则得到 DSC 为 20, CSC 为 100,缩放为 6,如表 5 所示.

表 5 生命周期评估分值

| 算法 | DSC | CSC | 缩放 |
|--------|-----|-----|----|
| 生命周期/年 | 10 | 50 | 3 |
| 分值 | 20 | 100 | 6 |

4.6 综合评估及解决方案

在以上每项评估中,都采用客观数据作为依据进行打分,从而达到最终结果更加客观的目的. 同时,现实场景中每一项产生的影响大小可能是不同的,在此基础上,为每项结果(x_i)添加一个权重参数(c_i),并通过调整权重参数来调整该项目在最终评估中的影响比例. 最终分数可计算如下:

$$s = \sum_{i=1}^5 c_i x_i, \quad \sum_{i=1}^5 c_i = 1 \quad (11)$$

在本文中,为简化模型,将所有的权重参数(c_i)设置为相等,即

$$c_1 = c_2 = c_3 = c_4 = c_5 = 0.2 \quad (12)$$

根据式(11)、式(12),可以得到如表 6 所示的最终综合分数. 最终决策规则非常简单,如果分数>50(最高分数 100 的 1/2),ASIC 是更好的选择;如果分数<50,则 FPGA 更好.

基于表 6 最终评估分值,可以得出结论如下. DSC (72>50)和 CSC (61.8>50)应在 ASIC 中实现,而缩放 (15.6<50)则应在 FPGA 中实现. 对于任何一种桥接算法,都可以用上述的方法进行评估,给出芯片资源的合理应用方案,为系统架构的早期设计提供可靠的依据.

表 6 综合评估分值

| 算法 | 权重 | DSC | CSC | 缩放 |
|--------|-----|-----|------|------|
| 市场应用度 | 0.2 | 50 | 100 | 12 |
| 性能 | 0.2 | 90 | 100 | 50 |
| 硬件成本 | 0.2 | 100 | 1 | 2 |
| 可参数化程度 | 0.2 | 100 | 8 | 8 |
| 生命周期 | 0.2 | 20 | 100 | 6 |
| 总分值 | 1.0 | 72 | 61.8 | 15.6 |

5 实验结果

本文提出的用于视频桥接的可编程异构芯片已在工业 22 nm 工艺中成功设计和实现. 整个芯片的版图如图 10 所示.

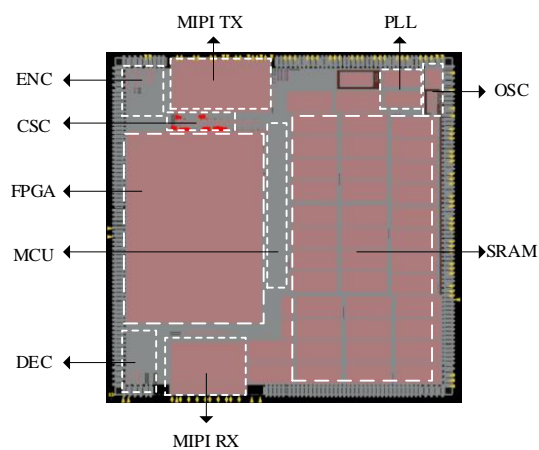


图 10 芯片版图概貌

封装后的最终芯片如图 11 所示,尺寸约为 4 mm × 4 mm. 与采用 28 nm 工艺的 AMD XC7K325T(其尺寸为 31 mm × 31 mm)相比,本文的芯片更适合移动显示领域的小型化应用.

图 1 中所列举的不同类型的视频桥接功能都可以



图 11 芯片外形与 XC7K325T 对比

在这个芯片上实现,并且无须外部存储器.图12展示了图1(d)中所描述数据流的应用演示.此演示模板将智能手机作为视频数据发送器.视频流数据格式为RGB,分辨率为 $1\ 080 \times 2\ 400$,刷新率为120 Hz,采用DSC压缩.它通过MIPI RX接口被接收到芯片中,数据通过DSI和DSC模块,然后进行从RGB到YUV的格式转换.接下来,在FPGA中按比例缩小,然后将数据存储在SRAM中.然后,数据格式被转换回RGB,并通过MIPI TX接口以 $1\ 080 \times 2\ 340$ 的分辨率和60 Hz的刷新率发送出去.最后,视频成功地显示在屏幕上.



图12 视频桥接应用演示

开发工具是一款国产工业电子设计自动化(EDA)软件.它支持从RTL设计到比特流下载的整个流程.所有ASIC功能模块都可以在RTL设计中实例化.资源使用情况报告如表7所示,其功能包括缩放、数据流控制、SRAM控制和其他粘合逻辑.

表7 FPGA资源利用率

| 资源种类 | 占用数量 | 可用数量 | 百分比/% |
|-----------|-------|-------|--------|
| LUT/s | 3 705 | 4 992 | 74.22 |
| REG/s | 1 810 | 9 984 | 18.13 |
| I/O pin/s | 9 | 45 | 20.00 |
| EMB18K/s | 4 | 16 | 25.00 |
| VMB8K/s | 10 | 16 | 62.50 |
| DSP9×9/s | 6 | 32 | 18.75 |
| PLL/s | 2 | 2 | 100.00 |

芯片的功耗测量结果如表8所示,整体功耗为196.306 mW.当尝试使用AMD XC7K325T实现此演示功能时,我们发现资源不足.因此,本文分别实现了每种算法,然后再进行相加.资源使用情况如表9所示.

表8 功耗测量结果

| 电源 | 电压/V | 电流/mA | 功耗/mW |
|------|-------|-------|---------|
| 数字电源 | 0.859 | 175 | 150.325 |
| 模拟电源 | 1.703 | 27 | 45.981 |

在这个演示中,没有使用DSC ENC.即使没有这个模块,总使用率也高达91.96%.与XC7K325T的约200 K LUT相比,本文的芯片只有5K LUT.就功耗而言,XC7K325T超过3 W,而本文的芯片约为200 mW.

表9 XC7K325T资源利用率

| 资源种类 | LUT/s | REG/s | LUT利用率/% |
|---------|---------|--------|----------|
| DEC | 118 652 | 30 165 | 58.22 |
| ENC | 86 170 | 19 745 | 42.28 |
| CSC | 4 688 | 1 990 | 2.30 |
| Scaling | 2 827 | 4 325 | 1.39 |
| MIPI | 23 538 | 15 784 | 11.55 |
| MCU | 40 931 | 21 089 | 20.08 |
| 合计 | 276 806 | 93 098 | 135.82 |

此外,本文尝试使用AMD Zynq Z7035来实现这个演示.由于Zynq内部有一个CPU,此架构中MCU的功能可以迁移到CPU上实现,因此MCU不占用逻辑资源.其他算法模块资源使用情况如表10所示,总共依然需要133.53%的资源.因此,Z7035也不能实现完整的演示功能.

Lattice Crosslink系列产品具有MIPI RX和TX接口,它们可以支持一些视频桥接的应用,CSC和图像缩放算法可以在FPGA中实现,但是DSC压缩解压缩算法太大,没有可以适配的FPGA容量,因此也不能实现此演示功能.

表10 Z7035资源利用率

| 资源种类 | LUT/s | REG/s | LUT利用率/% |
|---------|---------|--------|----------|
| DEC | 117 342 | 29 995 | 68.26 |
| ENC | 81 693 | 20 573 | 47.52 |
| CSC | 4 688 | 1 990 | 2.73 |
| Scaling | 3 097 | 4 587 | 1.52 |
| MIPI | 23 206 | 18 266 | 13.50 |
| 合计 | 230 026 | 75 411 | 133.53 |

这个应用程序可以通过ASIC来实现,但要在一个ASIC芯片上实现图1中提到的所有桥接应用并不容易.在商业市场上,ASIC厂商的解决方案是采用系列化的产品.例如,TI有SN65DSI83(支持单通道MIPI到LVDS转换)、SN65DSI84(支持单通道MIPI到双路LVDS转换),Toshiba有TC358767AXBG(支持MIPI到RGB转换)、TC358768AXBG(支持RGB到MIPI转换).其最高分辨率为 $1\ 920 \times 1\ 200$,60 Hz刷新率,其中最小的封装为 $4.5\text{ mm} \times 4.5\text{ mm}$,典型的工作功耗约为150 mW.由此可见,在面积和功耗方面,本文的芯片更接近ASIC,但却同时具备FPGA的可编程能力,一颗芯片可以具备多颗ASIC芯片的能力并具有灵活性.

本文的芯片可应用在更多的视频桥接应用场景

中,如图 13 中所示的墨水屏等.

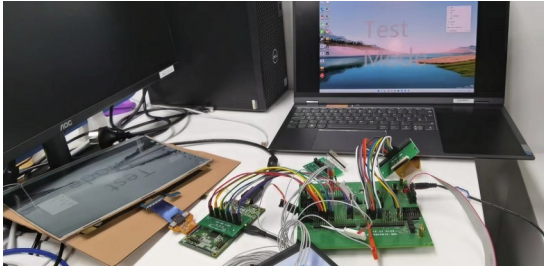


图 13 更多视频桥接应用场景

6 结论

本文提出了一种可编程异构芯片设计方法应用于视频桥接领域,它将 FPGA、MCU、ASIC 和 SRAM 集成在一个芯片中.并在分析视频桥接常用算法的基础上,提出了一种该芯片应用的评估方法和解决方案.该芯片已在 22 nm 工艺下成功流片,并实际应用落地.测试结果表明,在相同的视频桥接应用中,与传统的 FPGA 芯片相比,这种可编程异构架构可以获得更小的面积和更低的功耗.

参考文献

- [1] WANG Y Q, WANG C, ZHAO X Y, et al. A PCIe-based hardware acceleration architecture of the communication protocol stack[C]//2021 7th International Conference on Computer and Communications (ICCC). Piscataway: IEEE, 2021: 192-198.
- [2] BASLI A, KAHRAMAN C, ATES B, et al. Field programmable gate arrays based multi camera thermal and day vision video processing and distribution system[C]//2021 International Conference on Innovations in Intelligent Systems and Applications (INISTA). Piscataway: IEEE, 2021: 1-5.
- [3] ELNAWAWY M, FARHAN A, AL NABULSI A, et al. Role of FPGA in internet of things applications[C]//2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). Piscataway: IEEE, 2019: 1-6.
- [4] MIELCZAREK A, MAKOWSKI D, PEREK P, et al. Framework for high-performance video acquisition and processing in MTCA.4 form factor[J]. IEEE Transactions on Nuclear Science, 2019, 66(7): 1144-1150.
- [5] BEN AMARA A, PISSALOUX E, ATRI M. Sobel edge detection system design and integration on an FPGA based HD video streaming architecture[C]//2016 11th International Design & Test Symposium (IDT). Piscataway: IEEE, 2016: 160-164.
- [6] RANASINGHE N, BANGAMUARACHCHI R, SENEVIRATNE J, et al. SMPTE ST 2110 compliant scalable architecture on FPGA for end to end uncompressed professional video transport over IP networks[C]//2019 IEEE 30th International Conference on Application-specific Systems, Architectures and Processors (ASAP). Piscataway: IEEE, 2019: 235-238.
- [7] GEIER M, PITZL F, CHAKRABORTY S. GigE vision data acquisition for visual servoing using SG/DMA proxying[C]//Proceedings of the 14th ACM/IEEE Symposium on Embedded Systems for Real-Time Multimedia. New York: ACM, 2016: 1-10.
- [8] BAEK A R, LEE K, CHOI H. Speed-up image processing on mobile CPU and GPU[C]//2015 Asia Pacific Conference on Multimedia and Broadcasting. Piscataway: IEEE, 2015: 1-3.
- [9] ZHANG T, LIU X Y, WANG X D. High performance GPU tensor completion with tubal-sampling pattern[J]. IEEE Transactions on Parallel and Distributed Systems, 2020, 31(7): 1724-1739.
- [10] IGARASHI H, TAKANO F, MORIYOSHI T. Highly parallel transformation and quantization for HEVC encoder on GPUs[C]//2016 Visual Communications and Image Processing (VCIP). Piscataway: IEEE, 2016: 1-4.
- [11] ZHANG X F, MA Y, XIONG J J, et al. Exploring HW/SW co-design for video analysis on CPU-FPGA heterogeneous systems[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2022, 41(6): 1606-1619.
- [12] LIN Y C, ZHANG B Y, PRASANNA V, et al. HP-GNN: Generating high throughput GNN training implementation on CPU-FPGA heterogeneous platform[C]//Proceedings of the 2022 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. New York: ACM, 2022: 123-133.
- [13] HUANG Q J, WANG D Q, DONG Z, et al. CoDeNet: Efficient deployment of input-adaptive object detection on embedded FPGAs[C]//The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. New York: ACM, 2021: 206-216.
- [14] ZUCHOWSKI P S, REYNOLDS C B, GRUPP R J, et al. A hybrid ASIC and FPGA architecture[C]//IEEE/ACM International Conference on Computer Aided Design, 2002. Piscataway: IEEE, 2002: 187-194.
- [15] HO C H, YU C W, LEONG P H W, et al. Domain-specific hybrid FPGA: Architecture and floating point applications[C]//2007 International Conference on Field Programmable Logic and Applications. Piscataway: IEEE, 2007: 196-201.
- [16] JAIN A K, OMIDIAN H, FRAISSE H, et al. A domain-specific architecture for accelerating sparse matrix vector multiplication on FPGAs[C]//2020 30th International Conference on Field-Programmable Logic and Applications (FPL). Piscataway: IEEE, 2020: 127-132.
- [17] PARVEZ H, MARRAKCHI Z, MEHREZ H. Application specific FPGA using heterogeneous logic blocks[C]//Re-

configurable Computing: Architectures, Tools and Applications. Berlin: Springer, 2010: 92-109.

- [18] ZHAO R Z, NG H C, LUK W, et al. Towards efficient convolutional neural network for domain-specific applications on FPGA[C]//2018 28th International Conference on Field Programmable Logic and Applications (FPL). Piscataway: IEEE, 2018: 147-1477.
- [19] CUI G C, MENG T, SHI Y J, et al. Design of heterogeneous reconfigurable cipher engine basing on FPGA ASIC[C]//2019 5th International Conference on Big Data Computing and Communications (BIGCOM). Piscataway: IEEE, 2019: 261-265.
- [20] RAMACHANDRA C N, NAG A, BALASUBRAMONION R, et al. ONT-X: An FPGA approach to real-time portable genomic analysis[C]//2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). Piscataway: IEEE, 2021: 268-269.
- [21] KEFALAS N, THEODORIDIS G. Implementing VESA display stream compression encoder in FPGAs[C]//2019 29th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS). Piscataway: IEEE, 2019: 35-40.
- [22] 朱秀昌, 唐贵进. 视频显示流压缩的技术与标准[J]. 数据采集与处理, 2017, 32(3): 469-478.
ZHU X C, TANG G J. Video display stream compression technologies and standards[J]. Journal of Data Acquisition and Processing, 2017, 32(3): 469-478. (in Chinese)
- [23] Vedio Electronics Standards Association. VESA display stream compression (DSC) standard[S/OL]. (2017-01-18) [2024-05-26]. <https://glenwing.github.io/docs/VESA-DSC-1.2a.pdf>.
- [24] WALLS F, MACINNIS A S. 27.4L: *Late-news paper*: VESA display stream compression: An overview[J]. SID Symposium Digest of Technical Papers, 2014, 45(1): 360-363.
- [25] 彭春萍. 颜色空间转换在视频采集显示系统中的应用[J]. 工业控制计算机, 2012, 25(8): 22-23, 25.
PENG C P. Color space conversion in video sampling and display system[J]. Industrial Control Computer, 2012, 25(8): 22-23, 25. (in Chinese)
- [26] MAEDA Y, FUKUSHIMA N, HAMAMOTO T. Color transformation for compressive computing in image filtering[C]//2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (AP-SIPA ASC). Piscataway: IEEE, 2021: 88-92.
- [27] ITU-R Sector. Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios[S/OL]. (1995-10-01)[2024-05-26]. https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.601-6-200701-S!!PDF-E.pdf.
- [28] SURESH C, SINGH S, SAINI R, et al. A comparative analysis of image scaling algorithms[J]. International Journal of Image, Graphics and Signal Processing, 2013, 5(5): 55-62.
- [29] LIN C C, SHEU M H, CHIANG H K, et al. An efficient architecture of extended linear interpolation for image processing[J]. Journal of Information Science and Engineering, 2010, 26(2): 631-648.
- [30] MOON H M, KO K R, SHIN J, et al. A novel double linear-cubic convolution interpolation for digital image scaling[C]//Proceedings of the 29th Annual ACM Symposium on Applied Computing. New York: ACM, 2014: 1733-1734.
- [31] CHEN S L. VLSI implementation of a low-cost high-quality image scaling processor[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2013, 60(1): 31-35.
- [32] RAMESH G, PRASATH T A. An aphoristic study on different interpolation techniques for medical image scaling and its comparative analysis[C]//2021 International Conference on Computer Communication and Informatics (ICCCI). Piscataway: IEEE, 2021: 1-4.
- [33] 王莹, 王晶, 高岚, 等. 一种注意力机制优化方法及硬件加速设计[J]. 电子学报, 2023, 51(4): 1021-1029.
WANG Y, WANG J, GAO L, et al. An improved attention mechanism algorithm model and hardware acceleration design method[J]. Acta Electronica Sinica, 2023, 51(4): 1021-1029. (in Chinese)
- [34] BOUTROS A, YAZDANSHENAS S, BETZ V. You cannot improve what you do not measure: FPGA vs. ASIC efficiency gaps for convolutional neural network inference[J]. ACM Transactions on Reconfigurable Technology and Systems, 2018, 11(3):1-23.
- [35] KUON I, ROSE J. Measuring the gap between FPGAs and ASICs[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2007, 26(2): 203-215.

作者简介



王潘丰 男, 1980年10月出生于江苏省徐州市. 清华大学计算机科学与技术系创新领军工程博士生. 主要研究领域为智能卡、手机基带芯片、可穿戴芯片、FPGA芯片等.
E-mail: wpf20@mails.tsinghua.edu.cn



蔡懿慈 女, 1960年7月出生于天津市. 清华大学计算机科学与技术系教授、博士生导师. 主要研究领域为集成电路计算机辅助设计软件、算法及软件系统.
E-mail: caiyc@mail.tsinghua.edu.cn