

SOPC 技术在计算全息中的应用^①

郁伟勇¹, 简献忠¹, 奚利丰¹, 郭强², 於江贇¹

¹(上海理工大学 光电信息与计算机工程学院, 上海 200093)

²(国家卫星气象中心, 北京 100081)

摘要: 基于计算全息术的三维动态实时显示受到越来越多的关注, 而制约其发展的一大难题是其运算速度. 针对这一问题, 本文提出基于 SOPC(System On Programmable Chips)技术的计算全息硬件加速系统, 使多片 FPGA 硬件进行分块并行运算. 为了实现这一目标, 每片 FPGA 运算单元必须独立具备数据传输与计算全息算法加速两种功能. 在已有计算全息算法加速模块的基础上, 搭载 NIOS II 软核并移植 uC/OS II 操作系统及 LWIP 以太网协议栈. NIOS II 软核作为 FPGA 的主控, 控制计算全息算法加速模块及以太网口的数据传输, 实验结果证明该方法为实现计算全息三维动态实时显示提供了一种新的思路.

关键词: 计算全息; 以太网; FPGA; NIOS II 软核; uC/OS II; LWIP

Calculation of the Holography Based on SOPC Technology

YU Wei-Yong¹, JIAN Xian-Zhong¹, XI Li-Feng¹, GUO Qiang², YU Jiang-Yun¹

¹(University of Shanghai for Science & Technology, Shanghai 200093, China)

²(National Weather Satellite Center, Beijing 10081, China)

Abstract: More and more people pay attention to real-time three-dimensional dynamic display technology based on computer generated hologram. The major problem to restrict its development is computing speed. To solve the problem, we build PC-FPGA distributed computing holographic hardware acceleration system that multi-chip FPGA hardware block computing paralleled. Each chip FPGA unit must have the functions both of holographic algorithms computing and the data transmission to achieve this goal. Therefore, NIOS II soft core transplant uC/OS II operating system and LWIP Ethernet protocol stack based on already CGH algorithm acceleration module, NIOS II as the MCU to control CGH algorithm acceleration module and Ethernet data transmission. The article will introduce the NIOS II soft-core, uC/OS II and LWIP achieve to Ethernet data transmission process. To provide a new method to the realization of three-dimensional dynamic real-time display based on CGH.

Key words: CGH; Ethernet; FPGA; NIOS II; uC/OS II; LWIP

计算全息术由光学全息术发展而来, 其制作完全模拟光学全息的干涉过程并且抛开了光学全息术的诸多禁锢. 因此, 计算全息技术在光学领域中得到广泛应用. 1992 年, 日本千叶大学的研究小组开始开发专用图形硬件系统 (Holographic Reconstruction, HORN) 对 CGH 进行计算. 1993 年 3 月, 该研究小组完成了第一个硬件系统 HORN-1 的设计和制作. HORN-1 的设计采用的是并行处理模式的管线结构, 主要由 26 片

集成芯片搭建而成, 时钟频率为 10MHz, 运算速度大约是当时 PC 机的 100 倍, 国内也开展了许多关于计算全息三维立体显示技术的研究, 中国海洋大学物理系王金城教授利用数字合成彩虹全息技术实现了三维动态显示, 观察者可以在特定的角度观察到动态变换的三维立体图像. 山东师范大学的符秋丽提出一种基于 USB 接口的计算全息显示方案, 该方法是通过 USB 接口传输待显示的物波基本数据, 通过 DSP 等硬件来

① 基金项目:上海市研究生创新基金(JWCXSL1102);国家自然科学基金(40971200)

收稿时间:2012-12-10;收到修改稿时间:2013-01-11

实现高速计算全息编码和显示. 计算全息的运算速度是制约其发展的一大难题, 主要体现在两方面: 1) 计算全息图的制作需要相当大的数据量的计算, 现有的计算机系统无法满足实时处理的要求. 2) 在计算全息图制作的过程中, 会产生大量的图像数据信息, 在数据传输过程中要求接口能达到便捷、无失真及大批量传输的要求. 而目前 RS-232 接口、USB 接口等在传输速率及适用性上或多或少有些不足^[1].

针对以上问题, 提出以 FPGA 硬件作为计算单元对计算全息算法进行硬件加速的方法是通过构建一个 SOPC 硬件加速系统, 如图 1 所示, 即每一片 FPGA 都具备数据传输与计算全息算法硬件加速的完整功能. FPGA 硬件与主机之间通过以太网接口进行数据通信. 与其他嵌入式系统的串口通信接口相比, 以太网更加普及. 利用其使用广泛性可提供便捷的远程访问, 实现远程控制来满足特殊操作的需求. 此外, 经过适当剪裁和优化的 TCP/IP 协议栈, 也完全可以适应工业用途的需求. 以太网有使用度广泛、传输距离大、其布线线路成本低等优势.

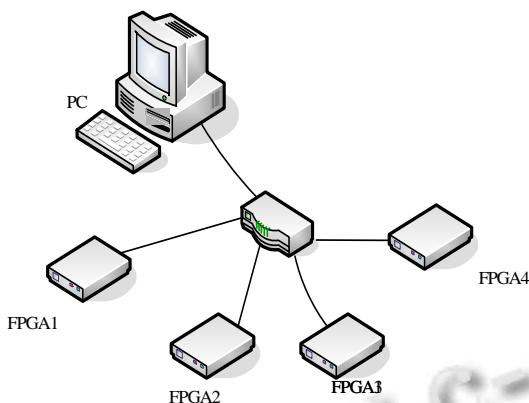


图 1 分布式计算全息硬件加速系统

1 分布式计算全息硬件加速系统原理介绍

图像的原始物点信息存储在 PC 机的存储器中, PC 机将物点坐标进行分配后, 通过以太网口将数据信息传送到各个 FPGA 硬件运算单元的 SDRAM 中. FPGA 搭载 NIOS II 软核并移植 uC/OS II 操作系统, 作为 FPGA 硬件运算单元的主控, 完成系统初始化, 管理以太网接口与上位机通信, 并获取这些坐标数据, 存入 SRAM 中. 计算全息运算单元自动从 SRAM 中获取原始物体的三维坐标离散值, 并自动开始计算相应物点的计算全息图. 计算单元自动将各个物点的计算全息图累加, 从而得到原

始物体的计算全息图的二进制数据, 再将 CGH 运算结果暂存在 SDRAM 中, 并通过以太网口将运算所得的数据传回 PC 机, 通过再现软件或光学方法还原物体图像.

1.1 计算全息原始物点数据分块传输原理

菲涅尔全息图的特点是记录平面位于物体衍射光场的菲涅耳衍射区, 物光由物体直接照射到全息平面上. 由于任意物体可以看作是由许多点光源组成的集合. 根据叠加原理, 物体的全息图可以看作是这些点光源的线性叠加.

基于计算全息图的光强度分布函数, 可以表示为:

$$I(x_h, y_h) = \sum_{b=1}^M A_b \cos \left[\frac{2\pi}{\lambda} \sqrt{(x_h - x_b)^2 + (y_h - y_b)^2 + z_b^2} \right] \quad (1)$$

其中, 参数 x , y 和 z 分别表示坐标的水平, 竖直和深度分量; 下标 b 和 h 分别代表物体和全息图.

经由泰勒级数展开及归一化处理后, 分别用 x_{hb} 和 y_{hb} 来代替 $(x_h - x_b)$ 和 $(y_h - y_b)$:

$$I(x_h, y_h) = \sum_{b=1}^N A_b \cos \left\{ 2\pi \left[\frac{pZ_b}{\lambda} + \frac{p}{2\lambda Z_b} (X_{hb}^2 + Y_{hb}^2) \right] \right\} \quad (2)$$

由于全息面上相邻点之间的距离远小于物体到全息面的距离, 可以用加法叠加运算来完成差值的计算, 从而得到其距离平方项的近似值. 这样就可以利用相邻点的计算近似得到当前行上所有点的光场分布. 将 (2) 式中的相位做单独列项处理可得:

$$\Phi_h = \frac{pZ_b}{\lambda} + \frac{p}{2\lambda Z_b} (X_{hb}^2 + Y_{hb}^2) \quad (3)$$

比较 Φ_{h+1} 和 Φ_h 的表达式, 可以得到二者的差值为:

$$\Phi_{h+1} - \Phi_h = \frac{p}{2\lambda Z_b} (2X_{hb} + 1) \quad (4)$$

比较 Φ_{h+1} 和 Φ_{h+2} 的表达式, 二者的差值为:

$$\Phi_{h+2} - \Phi_{h+1} = \frac{p}{2\lambda Z_b} [2(X_{h+1} - X_b) + 1] \quad (5)$$

由两次得到的差值表达式可以看出, 两个差值再相减等于常量, 记为 Δ . 记一阶差值 $\Phi_{h+1} - \Phi_h$ 为 Γ_0 , 则有:

$$\Phi_0 = \frac{pZ_b}{\lambda} + \frac{p}{2\lambda Z_b} (X_{hb}^2 + Y_{hb}^2) \quad (6)$$

$$\Gamma_0 = \frac{p}{2\lambda Z_b} (2X_{hb} + 1) \quad (7)$$

$$\Delta = \frac{p}{\lambda Z_b} \quad (8)$$

$$\Phi_{k+1} = \Phi_k + \Omega_k \quad \Omega_{k+1} = \Omega_k + \Delta \quad (9)$$

公式(6)(7)(8)主要计算各物点源在全息面每一行上第一个点的光强分布 $I(x_0, y_h)$, 然后根据公式(9)计算全息面同一行上其它点的光强分布 $I(x_0, y_{h+1}), I(x_0, y_{h+2}), I(x_0, y_{h+3}) \cdots I(x_0, y_{h+n})$.

由公式 1 全息平面上的光强分布公式可以看出, 在进行计算全息菲涅尔算法迭代运算前, 应先取得所要进行迭代的物体的坐标值. 假设任意物体有 N 个点光源组成, 第 b 个物点的位置为 (x_b, y_b) , 将这些物点的坐标值分别送入公式(1), 迭代出物点在全息面每一点上的光强分布. 而这 N 个点光源坐标即是物体的原始位置坐标. 由于最后所得的计算全息平面的光强度值是线性叠加的, 因此可以将物点原始坐标信息数据划分为 n 份分别进行计算.

通过 MATLAB 编程获取二值图像的灰度矩阵, 矩阵中的元素全部为 0 或 1, 其中 0 代表背景色, 1 为二维图像的有效信息. 那么, 获取二值图像的灰度矩阵之后, 将灰度矩阵中元素为 1 的元素所在的行数和列数分别作为采样点源的水平坐标和垂直坐标即 X 坐标和 Y 坐标. 在这里, 可以通过 MATLAB 编程获取二值图像的灰度矩阵并将矩阵中元素为 1 的元素所在的行数和列数输出, 这样一幅二值图像总共由 396 个像点组成, 关键 MATLAB 程序段如下:

```

For m=1:112
    for n=1:112
        if(objt(m,n)==1)
            row(i)=m;
            col(i)=n;
            i=i+1;
        end
    end
end

```

假如系统由 2 片 FPGA 硬件加速单元构成, 则将前 198 个数据点保存为 A1(灰度矩阵元素为 1 的行数), B1(将灰度矩阵元素为 1 的列数)的 EXCEL 表的形式送入第一片 FPGA. 同样, 将后 198 个数据保存为 A2、B3 送入第二片 FPGA^[2,3].

送入 FPGA 之后, 当数据传输完毕时, PC 机向计算全息图计算电路发送数据传输完毕的信号. 然后计算全息图计算电路的控制系统将暂存在 SDRAM 中的原始二维图像上第一点的 X 值和 Y 值赋值给 VHDL 程序中的基本处理单元, 并下达指令开始计算. 运算叠加得到的计算全息平面上每一点的光强度值保存在 FPGA 的 FIFO 缓存器中, 经由控制软件将光强度值通

过以太网口送入 PC 机. 等数据到达后, 送入再现还原软件 Holovision 将二进制数据转换成图像数据

2 系统硬件及 NIOS II 软核的搭建

2.1 系统硬件方案介绍

如图 1 所示, FPGA 硬件加速模块作为网络客户端通过 RJ45 端口经由网线连接在路由器上, PC 机作为网络服务端也通过网线连接在路由器上. 构成分布式计算全息加速系统的硬件平台.

FPGA 硬件加速模块具备数据传输与计算全息算法硬件加速的完整功能. 因此根据所需在 SOPC BUILDER 中构建 NIOS II 软核. 主要包括 NIOS II 处理器核、外部控制 PIO、Ethernet 接口(此处为 SPI 接口)、EPCS 控制器、系统时钟、Avalon 三态桥、计算全息计算单元等, 如图 2 所示. 外部器件与 NIOS II 软核相连, 构成完整的 FPGA 硬件加速模块^[4].

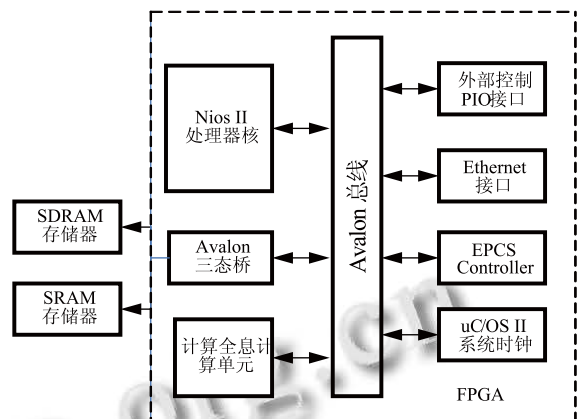


图 2 NIOS II 软核内部原理框图

2.2 计算全息运算单元设计

计算全息图算法的硬件电路以 FPGA 为核心, 在 FPGA 内部的计算全息图计算电路单元主要有基本处理单元 BPU 和 n 个加法运算单元 APU 组成, 如图 3 所示.

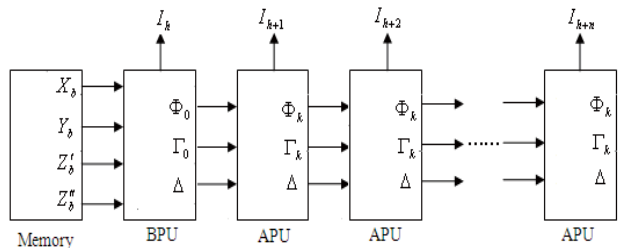


图 3 计算全息硬件电路计算单元方框图

BPU 用来实现公式(6)式、(7)式和(8)所计算的初始值;各个 APU 是用以实现(9)式的加法迭代计算.基本处理单元和加法处理单元的输出端根据其初始值得到全息图上各采样点的强度值.

2.3 网络传输模块设计

系统的网络芯片采用 Microchip 公司的 ENC28j60 芯片,带有行业标准外设接口(SPI)的独立以太网控制器.由 SOPC Builder 在 NIOS II 软核中生成两个 PIO 控制器, SPI 控制器,分别与 ENC28j60 芯片的片选端 LAN_CS、外部中断 LAN_nINT、及 SPI 口相连接. NIOS II 软核通过 Avalon 总线与各部件相连接,构成基于 SOPC 的系统架构.

3 以太网通信的实现

3.1 嵌入式实时操作系统 uC/OS II 的移植

uC/OS-II 是一种可移植、可裁剪、抢占式、实时多任务操作系统内核.它被广泛应用于微处理器、微控制器和数字信号处理器. ALTER 公司为 NIOS II 软核移植了 uC/OS II 操作系统,包括根据 NIOS II 的特点对数据类型重新定义,定义符合 NIOS II 特点的任务堆栈等^[5,6]. uC/OS II RTOS 的添加: NIOS II IDE 新建工程时,在“System Library Contents”的“RTOS”勾选 uC/OS II 选项,选择 SDRAM 作为程序段的存储地址. uC/OS II 各个选项的配置可通过“RTOS Option”进行选择 and 配置.

3.2 轻量级以太网协议 LWIP 的移植与实现

LWIP 实现的重点在保持 TCP 协议主要功能的基础上减少对 RAM 的占用,因此适合在资源不多的嵌入式系统中使用.

操作系统模拟层(sys_arch)提供了底层操作系统与 LWIP 之间的接口. sys_arch 需要为 LWIP 提供信号量(semaphores)和邮箱(mailboxes)两种进程间通信方式(IPC).因此,我们必须手动修改操作系统模拟层,使其实现与 uC/OS II 操作系统的结合.

在 sys_arch 中必须实现以下函数:

Void sys_init(void); 初始化 sys_arch 层.

Sys_sem_t sys_sem_new(); 建立返回一新信号量.

Void sys_sem_free(); 释放信号量.

Void sys_sem_signal(sys_sem_t sem); 发一信号.

U32_t sys_arch_sem_wait(sys_sem_t sem,u32_t timeout)等待指定的信号并阻塞线程.

这里, uC/OS II 中提供了相应的信号量参数,只需调用 uC/OS II 中建立信号量的相关函数,例如:

```
Sys_sem_t sys_sem_new(u8_t count)
{
    Return OSSemCreate(count);
}
```

同样的 sys_sem_free()对应 OSSemDel(), sys_sem_signal 对应 OSSemPost()等.

建立邮箱操作函数,邮箱在 LWIP 中用于消息传递.在移植 LWIP 的过程中,必须以 uC/OS II 的基础上重新实现.邮箱为结构体, sys_mbox_t 声明为指向该结构体的指针从而完成邮箱操作的构建工作.

```
Type struct lwip{
    OSEVENT* Mbox
    Void *pvaMsgs[MBOX_SIZE];
}LWIP_MBOX *sys_mbox_t; 在此基础上建立以下函数.
```

以下函数.

Sys_mbox_new() 建立一个新邮箱.

Sys_mbox_free() 释放邮箱.

Sys_mbox_post() 投递消息到指定邮箱.

Sys_arch_mbox_fetch()从指定邮箱接收消息.

3.3 网络驱动设备程序的添加

LWIP 的移植过程中很重要的一部分是将网络设备的驱动添加到 LWIP 协议中,它的成功添加与否关系到 LWIP 移植的成败.它提供了网络设备驱动添加的基本框架,即 ethernetif.c 文件.如图 4 所示,严格根据框架给出的顺序加入网络设备驱动.这一部分主要完成网卡的初始化,以网卡的接收,发送及中断服务程序^[7,8].

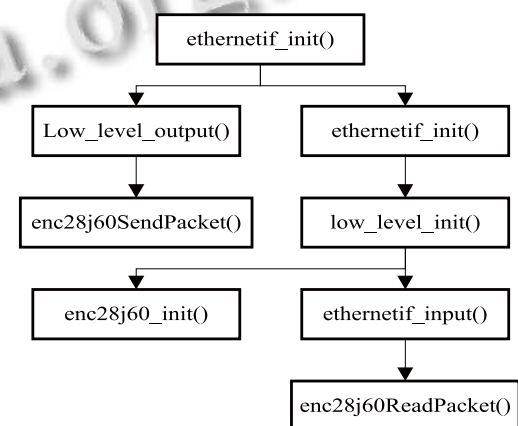


图 4 网络设备驱动添加实现顺序

(1) 网卡的初始化:

网卡的初始化是保证芯片正常运行的关键,对相关寄存器进行配置时应注意其先后顺序以及配置寄存

器之间的等待时间,加上延时函数.主要包括以下部分:对芯片进行软复位、设置芯片的工作模式、检测芯片、启动芯片等工作.

(2) Enc28j60 数据的接收:

数据接收包括查询接收中断、读取数据、释放缓冲区 3 个过程.首先,当设备驱动层(EMAC)接收到来自物理网络的数据包时,EMAC 产生外部中断通知主处理器进入中断服务程序 ISR,“接收数据包待处理中断标志位(EIR.PKTIF)”置 1,ISR 读取 EMAC 的接收状态寄存器获取当前的中断原因,如果确认无误,则立即向接收线程(即 ethernetif_input())发送信号量 OSSemPost(DataInput),接收线程收到信号量后,将 EMAC 接收缓冲区的数据取出并且释放接收缓冲区便于下一次接收.此时,数据包已由接收线程转换成 LWIP 的内部信息包 PBUF POOL 类型的 Pbuf 形式,送入网络层及传输层处理后传递给上层应用程序,完成接收过程.如图 5 所示.

(3) Enc28j60 数据的发送:

数据发送包括查询发送缓冲区,写数据到发送缓冲区,写命令发送数据.应用程序将数据送入 LWIP 进行分段,打包后送入发送线程即 low_level_output(),发送线程申请 EMAC 的使用权,如果网卡可用,立即占用并且堵塞后续申请.如果不可用,返回错误信息给应用程序.发送数据包,发送完成后产生中断,发送中断标志位置 1(EIR.TXIF),进入中断服务程序 ISR,ISR 读取发送状态寄存器获取当前的中断原因.若确认无误,则将数据写入 EMAC 的发送缓冲区送入网络进行数据传输.如图 5 所示.

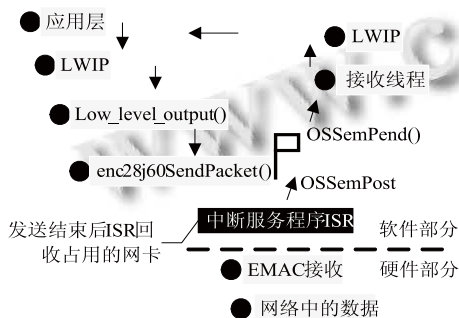


图 5 以太网数据收发流程

4 结语

针对计算全息运行速度慢的问题,本文提出构建 PC-FPGA 分布式计算全息硬件加速系统,使多片 FPGA 硬件进行分块并行运算的方法,详细介绍了 NIOS II 软

核的构建,及 uC/OS II 和 LWIP 实现以太网数据传输、对计算全息原始物点数据进行分块的相关流程.

将“上海理工”图片的原始坐标通过 Matlab 记录下来之后,送入两个模块分别进行计算,参数取 p=30 μm, λ=633nm, z_b=1500mm,如图 6 所示,所得结果为 Iout,结果以(00000,00000)开头,作为两段的区分.将从硬件模块接收到的波形数据分别保存,然后通过 MATLAB 程序将数据导入,生成计算全息图.原始图像及生成的计算全息图,如图 7 所示.

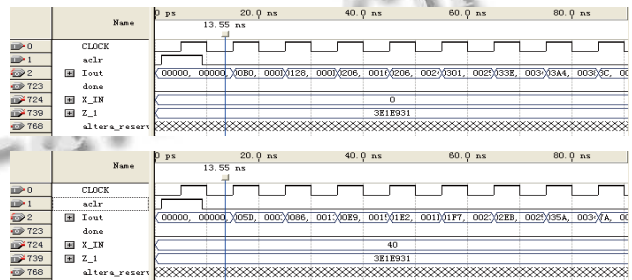


图 6 计算全息分块仿真



图 7 原始图像及生成的计算全息图

实验结果证明该方法为实现计算全息三维动态实时显示提供了一种新的思路.

参考文献

1 郭利辉,王淮生,简献忠,何玲玲.菲涅尔全息图的一种快速数值算法.微计算机信息,2010,26:176-177,191.
2 简献忠,张晗,范建鹏,周志刚.基于 FPGA 技术的计算全息研究.激光技术,2012,36(1):26-28,32.
3 简献忠,张晗,宋春茹,李菲.并行处理在计算全息中的应用.微计算机信息,2012,28(3):12-13,3.
4 李兰英.Nios II 嵌入式软核 SOPC 设计原理及应用.北京:北京航空航天大学出版社,2003.
5 任哲.嵌入式实时操作系统 uC/OS-II 原理及应用.北京:北京航空航天大学出版社,2005.
6 Labrosse JJ.嵌入式实时操作系统 uC/OS-II.北京:北京航空航天大学出版社,2003.
7 焦海波,刘健康.嵌入式网络系统设计.北京:北京航空航天大学出版社,2008.
8 ENC28J60 datasheet. Microchip Technology Inc. 2006.