文章编号:1671-4598(2020)12-0222-05 DOI:10.16526/j.cnki.11—4762/tp.2020.12.046 中图分类号:TP277 文献标识码:A

基于VP技术的星载智能计算机虚拟原型机的构建

吴兰蕙[1]，刘凯俊[1,2]，彭 攀[1,2]
(1.上海利正卫星应用技术有限公司，上海 201109；2.上海卫星工程研究所，上海 201109)

摘要：作为新技术和新生产力水平基础上涌现出来的产物，以全新设计理念研制的现代小卫星相较于传统大卫星在功能密度上有了显著提升，得到了广泛的应用；针对目前现代小卫星技术面临的设计复杂度提高和研发周期缩短的双重压力，提出了一种基于虚拟原型（VP）技术来构建星载智能计算机的虚拟原型机的设计，利用SystemVerilog的直接编程接口（DPI）技术在仿真验证平台上对C语言的调用，实现不同硬件平台间的通信，模拟星载计算机在轨运行的数据流向；该设计能缩短卫星的研发周期，降低卫星研制的成本；通过工程实例中虚拟原型机中的构建流程展示，结果证明，基于VP技术构建的虚拟原型机能够模拟星载计算机的部分工作流程，实现星载计算机的功能仿真，满足了商业小卫星快速迭代的研发需求，为小卫星的研制提供新的思路。

关键词：星载智能计算机；虚拟平台；SystemVerilog DPI接口

Virtual Prototype Design of On—Board C&DH Based on Virtual Platform

Wu Lanhui[1], Liu Kaijun[1,2], Peng Pan[1,2]
(1. Shanghai Lizheng Satellite Application Technologies Co., Limited, Shanghai 201109, China;
2. Shanghai Institute of Satellite Engineering, Shanghai 201109, China)

Abstract: As a product emerging on the basis of new technologies and new high—performance levels, modern small satellites developed with new design concepts have significantly improved their functional density compared with traditional large satellites and have been widely used. Aiming at the dual pressure of increasing design complexity and shortening R&D cycle faced by modern small satellite, a virtual prototype design of on—board C&DH based on virtual platform (VP) is proposed. The communication between different hardware platforms is realized virtually in simulation by the calls of C language using direct programming interface (DPI) of SystemVerilog, so that the data flow of on—board C&DH is simulated. The design can shorten the R&D cycle of the satellite and reduce the cost of satellite development. Through the demonstration of construction of virtual prototype, the results prove that the virtual prototype can simulate part of the workflow of the on—board C&DH and realize the functional simulation. The design satisfies the demand of rapid iteration of commercial satellites and provides new ideas for the development of small satellites.

Keywords: on—board C&DH; virtual platform; system Verilog DPI interface

0 引言

目前，现代小卫星作为“新航天”浪潮的重要组成部分，已进入新的发展阶段[1]。从近年来小卫星的发展来看，一方面在微电子技术飞速发展的支持下，小卫星可以以较低的单星成本进行批量化制造；另一方面，小卫星能以一箭多星、空中发射及在轨弹射等手段快速实现小卫星星座的批量部署，降低了进入空间成本。成本低廉、发射灵活、高效费比的小卫星通过星座组网和优化轨道设计组成小卫星星座，既保持了小卫星的原有优势，又可以获得与大卫星相媲美的能力。为了满足未来大规模商业小卫星星座部署的需求，需要在能批量生产商业小卫星的同时，控制单颗卫星的成本，缩短卫星的研发周期。并且，随着卫星数量的急速增加和卫星获取数据能力的增强，未来卫星将面临海量的数据待处理，迫切要求卫星数据处理的自动化和智能处，提高数据处理的能力，即通过卫星高性能计算能力的增强来实现大数据智能分析的功能。因此，小卫星的设计目前面临设复杂度的提高和研发期缩短的双重压力。

为了适应小卫星快速迭代的发展需求，本文提出采用虚拟原型VP技术（VP，virtual platform）构建星载智能计算机（C&DH，command & data handling）的虚拟原型机来加速商业小卫星的研发流程[2]。VP技术是软硬件协同开发技术发展的产物，即在没有物理硬件的情况下搭建一个虚拟的仿真平台，通过编程接口让软件开发者能更早地模拟在目标硬件模型上的编程环境，调试自己的程序，同时硬件设计也能及时收到反馈，作为硬件优化的参考。考虑到航天器设计前期研发难度大，一旦投产后再更改设计成本大等因素，本文通过构建虚拟原型机来模拟异构星载计算机的数据交互，通过仿真进行软件算法设计，及时优化软硬件设计，使目标星载智能计算机具备一步正样的能力，提高效费比。

收稿日期:2020-06-08； **修回日期**:2020-07-18。

作者简介:吴兰蕙(1994-)，女，江苏苏州人，硕士生，工程师，主要从事星载图像处理方向的研究。

1　虚拟原型机的设计与实现

1.1　虚拟原型机的设计原理

商业小卫星普遍采用不同计算架构的商用 COTS 器件（COTS，commercial off－the－shelf）实现异构星载计算机的硬件设计，如 ARM＋FPGA＋DSP 结构。随着卫星智能化的发展趋势，ARM＋FPGA＋GPU 的异构硬件设计也涌现出来。针对目前小卫星的跨平台异构硬件设计，各平台之间没有统一的验证语言能直接进行跨平台仿真。本文提出的基于 VP 技术的虚拟原型机以 FPGA 的仿真平台为核心，通过 SystemVerilog 的直接编程接口（DPI，direct programming interface）调用 C 语言实现跨平台数据交互的功能。各软件平台虽然无法互通，但通常都提供自己的 C 语言的接口库，可以通过 C 语言接口实现跨平台的间接调用。针对卫星而言，卫星在轨运行的实际数据流如图 1 所示，本文提出的虚拟原型机模拟的数据流主要为图 1 中的虚线部分，即有效载荷和星载智能处理机的数据交互以及星载智能处理机内部的数据流向。

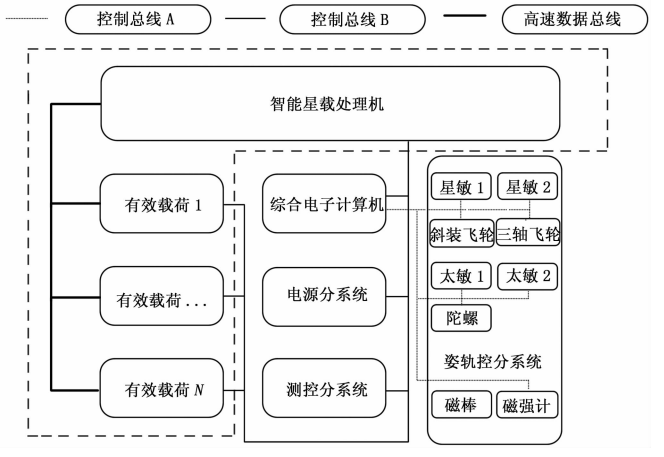


图 1　卫星在轨运行数据流图

虚拟原型机中主要使用的语言为 C 和 SystemVerilog。Verilog 是航天硬件设计中的常用语言，SystemVerilog 则是由 Verilog 语言发展来的硬件描述、硬件验证的统一语言，是 Verilog 语言的高层次扩展和增强，目前已被采纳为 IEEE 标准，并获得了主流电子设计自动化工具供应商的支持。相较于 Verilog 的编程语言接口（PLI，programming language interface）与其它编程语言交互的复杂的调用方式，SystemVerilog 的 DPI 接口能通过简单的 import 声明就可以像直接调用 SystemVerilog 的内部函数一样调用 C 或 C＋＋的函数功能。C 或 C＋＋语言包含很多库函数，且实现复杂模型比通过 HDL 语言实现要更为简单，开发人员可以直接在 C 环境下写 C 测试用例，降低验证测试的难度，同时，C 语言的执行速度也更快，可以提高仿真速度。此外，C 语言环境移植方便，支持的平台也更多，也为测试代码的复用提供了基础[3-5]。基于 DPI－C 技术用 C 语言实现 Socket 编程，基于 TCP/IP 协议，或 UDP 协议可以实现跨平台的进程间通信，方便模拟异构硬件设计的跨平台数据交互[6]。通过 DPI－C 技术可以快速搭建仿真平台，构建原型机，提高仿真效率。

1.2　虚拟原型机的技术实现

虚拟原型机的 FPGA 架构仿真主要依托于专业的 FPGA 仿真软件，这里使用的是 Mentor Graphics 的多语言 HDL 仿真环境 ModelSim，是业界唯一的单内核支持 VHDL 和 Verilog 混合仿真的仿真器，既可独立仿真使用，也可与其它工具软件联合仿真使用，如 Xilinx 的 Vivado 和 Intel 的 Quartus，方便后续软件移植到硬件平台后的联合仿真。

模拟星载数据流通过 DPI－C 技术编写 C 代码读取数据流信息，其中，C 代码必须包含头文件 svdpi.h。该头文件包含了 C 语言和 SystemVerilog 语言数据类型的映射关系，具体如表 1 所示。DPI 接口编译的时候并不会检查数据类型的兼容性，需要使用者根据数据映射关系自行保证数据匹配的正确性[7]。

表 1　SystemVerilog 和 C 数据映射表

SystemVerilog	C(input)	C(output/inout)
byte	char	char *
shortint	short int	short int *
int	int	int *
longint	long int	long int *
shortreal	float	float *
real	double	double *
string	const char *	char * *
string[n]	const char * *	char * *
bit	svBit(unsigned char)	svBit * (unsigned char)
logic，reg	svLogic	svLogic *
bit[N:0]	const svBitVecVal *	svBitVecVal *
reg[N:0]/logic[N:0]	const svLogicVecVal *	svLogivVecVal *
OpenArray[ ] (import only)	const svOpenArray Handle	svOpenArray Handle
chandle	const void*	void*

在 SystemVerilog 代码中调用 C 语言实现的函数或任务，其定义如下：

import "DPI－C" [c_identifier ＝][pure][context] function type name(args);

import "DPI－C" [c_identifier ＝][context] task type name (args);

该函数具体内容在 C 语言中进行定义。

在 SystemVerilog 代码中导出函数或任务，其定义如下：

export "DPI－C" [c_identifier ＝]function type name;

export "DPI－C" [c_identifier ＝]task type name;

在 SystemVerilog 中进行函数或任务的定义后，使用上述语句导出，在 C 语言中进行 extern 申明后，可在后续程序中直接使用该函数。通过 DPI－C 技术，可以根据使用的具体情况由 SystemVerilog 端和 C 语言端分别进行处理。经过 DPI 接口调用后，通过 ModelSim 混合编译 SystemVeril-

og 和 C 语言，即可运行。

异构平台数据交互的仿真通过 DPI 接口使用 C 语言进行 Socket 编程基于 TCP/IP 协议实现网络传输功能，如创建 ServerSocket 和 ClientSocket，打开 Socket 链接，按照协议进行 Socket 读/写操作，关闭 Socket 等基本操作，并进行封装。具体包括：

socket_init()：初始化，仅开头调用一次；

socket_shutdown()：关闭所有链接模块，仅结束调用一次；

socket_open(input string uri)：建立端口连接，uri 为 TCP Socket 链接，如 tcp://hostname:port；

socket_close(input chandle handle)：关闭连接，handle 为 socket_open 返回的句柄；

socket_send(input chandle handle, input string data)：向 handle 发送数据；

socket_receive(input chandle handle)：从 handle 接收数据。

SystemVerilog 语言中通过 DPI 接口可以直接导入上述函数进行调用，同时，在另一端建立 TCP/IP 的端口即可实现与 FPGA 的基于 TCP/IP 协议的双向传输。

2 工程实例

本文课题研究的最终目标是设计商业小卫星的星载智能计算机实现卫星在轨进行舰船实时检测的任务。该星载智能计算机采用 ARM+FPGA+GPU 的异构硬件设计，其结构如图 2 所示。该星载智能计算机由两大模块组成：以 FPGA 为主的图像预处理模块和以 GPU 为核心的神经网络计算模块。模块间通过 PCIE 总线进行数据交互。其中，FPGA 模块分为 PS，PL 两个部分，PS 部分包含 ARM 处理器，负责流程处理，任务调度等功能。PL 部分负责数据的预处理，流水线等功能。其数据流向为从卫星载荷相机获得的图像数据流输入首先进入 FPGA 模块进行图像预处理工作，后将预处理后的数据通过 PCIE 总线传送至 GPU 计算模块进行舰船检测，处理后的结果再由 GPU 模块送回至 FPGA 模块，最后由 FPGA 将得到的结果通过 LVDS 或 RS422 送回至星务系统。

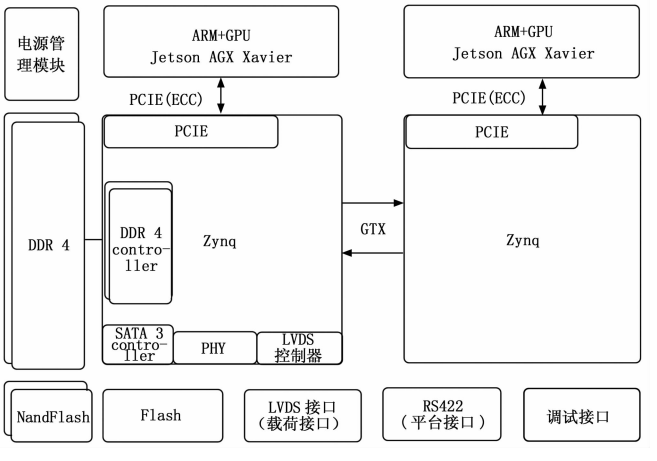


图 2 智能星载计算机硬件设计图

针对智能星载计算机的硬件设计与数据流向，本文提出的虚拟原型机进行一一映射构建，其映射关系如图 3 所示。GPU 模块进行的舰船检测算法仿真通过上位机在同一深度学习框架下实现，方便后续硬件平台上代码的移植。FPGA 模块由 ModelSim 进行仿真实现。其中，FPGA 的 PL 端图像预处理过程可以根据实际片上资源由 Verilog 语言实现。FPGA 的 PS 端控制仿真需要考虑芯片系统级别上的外别访问行为，由真实的硬件设备进行仿真存在诸多限制和不便，例如针对不同子系统需要预留不同接口进行调整，以及没有纯软件仿真方便。通过 DPI－C 技术可以构建虚核来实现虚拟处理器的设计，同时满足总线接口读写，中断响应等处理器特性。PICE 实现的数据传输简化为网口传输实现。通过 DPI 接口调用 C 语言可以简单地通过 Socket 编程在 FPGA 平台建立 TCP/IP 通信协议，实现不同模块间的数据互通。FPGA 硬件仿真中，无法直接输出结果的问题也可以由 DPI 接口将函数导出到 C 语言一侧进行打印输出，方便调试。

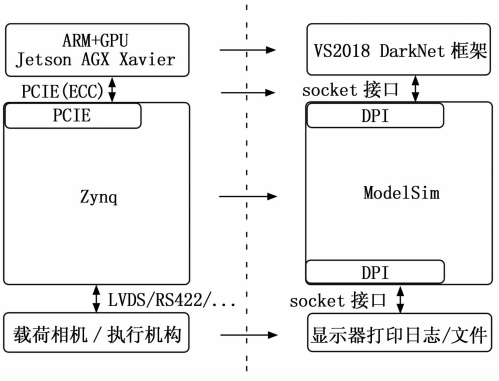


图 3 智能星载计算机与虚拟原型机关系映射图

根据映射关系，本文提出搭建的虚拟原型机模拟的实现过程如下：首先，通过 DPI－C 接口读取图像数据流，之后在 ModelSim 仿真软件上实现 FPGA 的图像预处理功能，最后通过 Socket 编程实现的 TCP/IP 通信协议建立 FPGA 模块与上位机的数据互通，由上位机加载训练好的神经网络算法进行目标检测，最终将结果送回至仿真平台并对图像数据进行可视化处理。

其中，本节展示的工程实例对实际使用的算法进行简化，仅通过仿真过程中基本功能的实现来模拟虚拟原型机的数据流向，实际工程与仿真模拟的软件流程如图 4 所示。具体实现过程如下：

1）实例选择的图片输入来源于航拍图像数据集 DOTA (a large－scale dataset for object detection for object detection in aerial images)[8-9]。选取该数据集中一张带有舰船的图像用作实例展示，通过 DPI 接口实现数据输入。针对 DPI－C 接口中数据传递的数据类型，这里选择以 OpenArray 的形式进行传递。以该形式进行数据传递时，可以方便 SystemVerilog 代码直接对数据的长度进行控制，避免了当数据大小发生变化时需要在 C 代码中反复进行更改的情况。实际过程中，FPGA 可以直接从相机获得 RAW 格式数据得

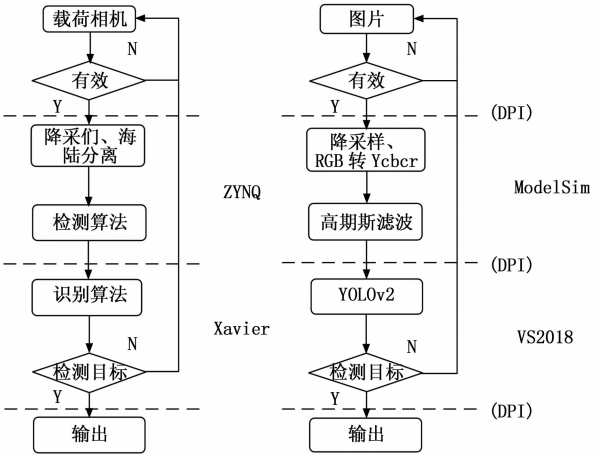


图 4　软件流程图

到 RGB 图形数据。为方便后续算法移植到硬件平台，这里不直接读取图片，通过 C 语言将其转为 FPGA 可读取的文件形式，并保存 RGB 三通道的图像数据，由 FPGA 通过 DPI 接口调用 C 程序实现图像数据的读取过程。

2）实现数据输入后，通过 Verilog 进行 FPGA 模块图像预处理的仿真过程，相较于实际工程应用在这里进行了相应的简化。首先，为提高检测速度，对原始图片进行降采样和切片，得到尺寸为 800 * 600 的图片。然后，将 RGB 色彩空间转为 *YCbCr* 色彩空间，首先通过高位补低位先将 RGB565 转化为 RGB888，再由公式计算转为 *YCbCr*。其中涉及到的浮点运算可以通过先将数值扩大 256 倍再右移 8 位转化为 FPGA 的无浮点乘法和加法运算，更改后的色彩空间转换公式如下：

$$Y = ((77 * R + 150 * G + 29 * B) >> 8);$$

$$Cb = ((-43 * R - 85 * G + 128 * B) >> 8) + 128;$$

$$Cr = ((128 * R - 107 * G - 21 * B) >> 8) + 128;$$

之后通过三级流水线设计进行 Verilog 代码实现。其中 *YCbCr* 色彩空间下的 *Y* 通道代表了颜色的亮度部分，单独提取 *Y* 通道分量得到的图像可视为灰度图。对获得的灰度图进行 FPGA 下的高斯滤波处理。实际应用中，为提高检测精度，针对舰船识别的应用背景，图像预处理过程更加复杂，除去常规平滑、滤波等预处理步骤外，还需进行去云雾处理、海陆分离等预处理过程。后续算法移植过程中，综合考虑 FPGA 与 GPU 的性能和功耗问题，部分检测网络可以放在 FPGA 上实现。针对检测网络的常用模块，如卷积层、池化层等，其 FPGA 上的实现思路与高斯滤波的实现思路类似，由不同大小的卷积核按照步长滑过图片进行卷积运算实现。考虑到 FPGA 处理的并行性特征，可以通过行缓存（LineBuffer）构建 IP 核实现 FPGA 上的图像卷积运算[10]。利用 FPGA 内部的 Block Ram 缓存若干行的图像数据，再以并行流水线设计进行卷积运算。Verilog 实现卷积运算的仿真波形如图 5 所示。

3）将预处理后的图像数据进行分包，通过 Socket 编程实现的 TCP/IP 协议网口传输至模拟 GPU 模块加载神经网

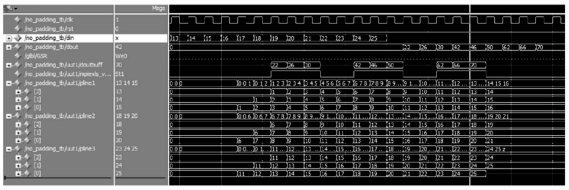
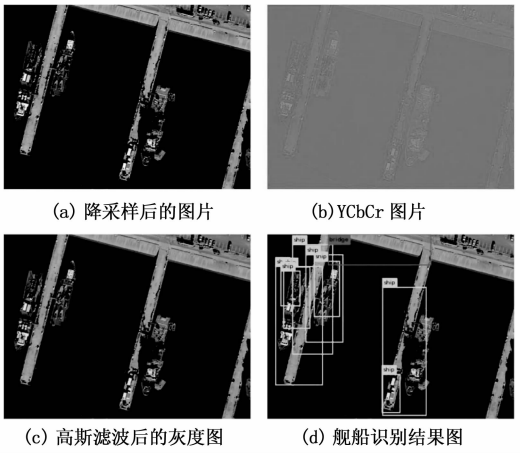


图 5　波形示意图

络模型进行目标检测。TCP/IP Socket 由 C 代码实现，包含打开/关闭 Socket，进行读/写等基本操作，FPGA 端通过 DPI 接口调用并进行联合编译。在上位机建立 TCP/IP Socket 端口，与 Modelsim 模拟的 FPGA 模块进行连接，并接收图像数据。本实例中虚拟原型机中的模拟 GPU 模块为上位机训练的深度学习网络。这里使用 Darknet 框架下的 YOLOv2 算法进行舰船检测任务[11-12]。YOLO 算法是常用的基于回归的目标识别算法，识别速度快，并且其原生的 C 语言版本 Darknet 框架较为轻型，无任何依赖项，可移植性高，易于在不同平台上进行实例展示。实际工程中，会根据舰船检测的特殊性对 YOLO 算法进行优化，进一步提升舰船识别的检测精度，降低误检率。通过 YOLO 模型识别后得到的检测结果同样经由 TCP/IP Socket 传输送回 FPGA 模块。FPGA 模块经由 DPI 接口导出打印功能，由 C 语言输出结果进行展示。

3　实验结果分析

上述工程实例展现了整个虚拟原型机的构建过程，通过与硬件架构一一映射的纯软件虚拟原型机的仿真结果，可以对硬件架构的设计的可行性提供反馈。本文构建的虚拟原型机模拟实现的星载智能计算机进行舰船识别的仿真生成的图像结果如图 6 所示。其中，图 6（a）为原始图像经过降采样和裁切之后的得到的 800 * 600 尺寸三通道的 RGB 图像；图 6（b）为 FPGA 下经由色彩空间转换后的得到的 YCbCr 图像；图 6（c）为提取 Y 分量后得到的灰度图再进行高斯滤波后得到的图像；图 6（d）为 YOLO 算法进行舰船识别后得到的检测结果图。



(a) 降采样后的图片　(b) YCbCr 图片

(c) 高斯滤波后的灰度图　(d) 舰船识别结果图

图 6　图像结果图

结果图的图像数据无损失说明了虚拟原型机中的数据交互功能可以正常运行，通过 DPI 接口建立的 TCP/IP 协议可以实现 FPGA 平台与其它平台数据的双向传输。同时，结果图中的降采样图、YCbCr 图、高斯滤波后的灰度图与理论上的图片结果一致，验证了硬件设计中由 FPGA 进行图像预处理设计的可行性。由于 FPGA 的并行流水线设计，其延时是固定的，方便预估检测算法的实时性。舰船识别后的结果图在没做其它算法优化的前提下能成功识别出部分目标，对检测算法的可行性进行了验证。可以在无硬件环境下实现神经网络算法的整体设计，同时，也可以对仿真环境下检测性能的分析来指导硬件设计中不同算力的 GPU 选型。

由此可见，该虚拟原型机在脱离硬件平台的情况下能成功模拟星载智能计算机的数据流向并仿真实现星载智能计算机的部分功能。并且，通过 DPI 接口调用 C 程序实现了跨平台间的数据交互，简化了虚拟原型机的搭建过程，易于实现。通过虚拟原型机的仿真结果可以验证星载计算机的硬件架设计，及时对硬件设计进行优化，使卫星具备一步正样的能力，降低卫星研制的成本。同时，在虚拟原型机上提前进行软件设计，可以缩短卫星的研制周期。虚拟原型机与星载智能计算机一一映射，方便后续不同架构间的算法移植。

4 结束语

优化卫星研制周期，降低成本是商业小卫星快速发展需要重点关注的问题。本文提出的基于 VP 技术构建的星载智能计算机虚拟原型机可以仿真智能星载计算机的部分功能，模拟其内部的数据流向，为硬件设计的开发提供了反馈，使卫星具有一步正样的能力，提高了卫星研制的效费比。同时，可以在虚拟原型机上提前进行软件算法设计，缩短了卫星研制的时间，优化了卫星的开发周期。并且，针对目前智能化卫星的异构硬件设计，通过 DPI 接口调用 C 程序简化了跨平台间的数据交互，降低对开发人员的能力要求。

后续，虚拟原型机还可通过 DPI－C 技术构建虚拟多核处理器来模拟 ARM 处理器的控制部分，实现星载智能计算机更多功能的仿真。针对本课题，后续将在虚拟原型机上分别对 FPGA 模块的预处理算法和 GPU 模块的目标识别算法进行优化设计，提升舰船检测的准确率，达到预计性能指标。最后再将算法移植到实际硬件的星载计算机板进行融合，完成舰船实时检测的星载智能计算机的研制。

参考文献：

[1] 唐 亮，刘鸿鹏，何慧东. 全球小卫星现状及发展 [J]. 国际太空，2019 (6)：36－41.

[2] 李连军. 航天器功能行为虚拟原型建模方法及实现技术研究 [D]. 长沙：国防科学技术大学，2006.

[3] 虞致国，魏敬和. 基于 SystemVerilog DPI 的 ARM SoC 虚拟调试验证平台的设计 [J]. 微电子学与计算机，2009，26 (11)：117－119，123.

[4] 李 璐，周春良，冯 曦，等. 基于 DPI－C 接口的可扩展 SOC 验证平台 [J]. 电子设计工程，2018，26 (4)：136－140.

[5] 祝周荣，关俊强，李前进，等. SVDPI 技术在 FPGA 仿真验证的应用探讨 [J]. 计算机测量与控制，2018，26 (6)：264－267.

[6] 熊世桓. 用 Socket 编程实现 TCP/IP 网络接口 [J]. 贵州教育学院学报 (自然科学)，2003 (4)：86－90.

[7] Synopsys. Reference verification methodology user guide [Z]. 2004.

[8] Xia G S，Bai X，Ding J，et al. A large－scale dataset for object detection in aerial images [A]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition [C]. 2018：3974－3983.

[9] Ding J，Xue N，Long Y，et al. Learning RoI transformer for oriented object detection in aerial images [A]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) [C]. 2019：2844－2853.

[10] 朱学亮，柴志雷，钟传杰，等. 基于 FPGA 的图像卷积 IP 核的设计与实现 [J]. 微电子学与计算机，2011，28 (6)：188－192.

[11] Redmon J，Farhadi A. YOLO9000：better，faster，stronger [J]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition [C]. 2017：7263－7271.

[12] 马 啸，邵利民，金 鑫，等. 改进的 YOLO 模型及其在舰船目标识别中的应用 [J]. 电讯技术，2019，59 (8)：869－874.

(上接第 221 页)

[8] 蔡晓丽，钱 诚. 基于改进的粒子群算法的云资源调度策略 [J]. 微电子学与计算机，2018，35 (6)：28－30.

[9] Mathiyalagan P，Dhepthie U R，Sivanandam S N. Grid scheduling using enhanced ant colony algorithm [J]. ICTACT Journal on Soft Computing，2010 (2)：400－407.

[10] Srikanth U，Maheswari V U，Shanthi P，et al. Tasks scheduling using ant colony optimization [J]. Journal of Computer Science，2012，8 (8)：1314－1320.

[11] Pandey S，Wu L，Guru S M，et al. A particle swarm optimization－based heuristic for scheduling workflow applications in cloud computing environments [R]. Technical Reports. University of Melbourne，2009.

[12] Feng M，Wang X，Zhang Y，et al. Multi－objective particle swarm optimization for resource allocation in cloud computing [A]. Proceedings of IEEE CCIS [C]. 2012：1161－1165.

[13] Xiong Y H，Huang S Z，Wu M. A Johnson's－rule－based genetic algorithm for two－stage－task scheduling problem in data－centers of cloud computing [J]. Journal of Latex Class Files，2014，13 (9)：1－14.