

面向机载应用的领域专用加速器研究

王荣阳, 曲国远, 童 歆, 陈 昊, 李 威, 徐佩园

(中国航空无线电电子研究所, 上海 200233)

[摘 要] 机载环境下数据处理规模的剧增以及人机混合智能的应用使得传统的以 CPU 为核心计算单元的架构已不能满足计算需求。在满足延时、精度等指标的情况下, 选用高能效的处理器或处理器组合来快速准确地处理这些数据成为机载计算领域面临的重要问题。按照常规处理器、领域专用加速器两大类型对各自主要代表性处理器的架构特点进行了分析和总结, 得出了各类处理器在机载情况下的主要适用场景和应用情况。根据领域专用的设计思想开发了面向数据关联应用的专用加速器, 对数据关联算法中的统计距离计算和分簇处理这两个计算瓶颈进行了定制化的加速设计, 并在基于 FPGA 的平台上进行了测试验证, 结果表明, 加速器对于统计距离计算的加速效果约为 FT2000/4 单核性能的 10 倍, 对于分簇处理的加速效果约为 FT2000/4 单核性能的 3 倍, 整体运算速度相比 FT2000/4 处理器的单核提升了 5 倍。

[关键词] 机载计算平台; 异构计算; 领域专用; FPGA; 专用加速器; 指令集; 微架构; 数据关联

[中图分类号] V247.1

[文献标识码] A

[文章编号] 1006-141X(2022)03-0001-08

Research on Domain Specific Accelerators for Airborne Applications

WANG Rong-yang, QU Guo-yuan, TONG Xin, CHEN Hao, LI Wei, XU Pei-yuan

(China National Aeronautical Radio Electronics Research Institution, 200233 Shanghai, China)

Abstract: The rapid increase in the scale of data processing and the application of human-computer hybrid intelligence have made the traditional architecture with CPU as the main computing unit no longer able to meet the computing needs in the airborne environment. In order to satisfy indicators such as delay and accuracy, selecting a high energy efficiency processor or combination of processors to process these data quickly and efficiently has become an important issue in the field of airborne computing. The architecture features, applicable scenarios, and application conditions of the respective main representative processors are analyzed and summarized, according to the general-purpose and special-purpose types. A dedicated accelerator for data-related applications has been designed and implemented on an FPGA-based platform. The results show that the acceleration effect of the accelerator for statistical distance calculation is about 10 times that of FT2000/4 single-core performance, and the acceleration effect for clustering processing is about 3 times that of FT2000/4 single-core performance. The overall computing speed is increased by 5 times.

Key words: airborne computing platform; heterogeneous computing; domain specific; FPGA; specific accelerator; instruction set; micro architecture; data association

收稿日期: 2021-09-05

引用格式: 王荣阳, 曲国远, 童歆, 陈昊, 李威, 徐佩园. 面向机载应用的领域专用加速器研究 [J]. 航空电子技术, 2022, 53(3): 01-08.

近年来, 机载智能计算、数据处理技术得到快速发展, 各类算法的复杂度、运算规模不断增加, 这对承载算法运行的硬件平台提出了越来越高的要求^[1]。机载计算平台主要面临以下几个方面的挑战: (1)数据规模大: 雷达信号处理、合成孔径雷达(SAR: Synthetic Aperture Radar)图像处理领域, 数据吞吐量达 100 Gbps 以上^[2-3]; (2)算法复杂度高: 传感器信号处理算法增加各类滤波处理、人工智能算法计算量大, 参数众多^[4]; (3)精度要求高: 数字信号处理、空战博弈、辅助决策等应用中为了获得高精度, 要求单/双精度浮点运算; (4)实时性要求高: 空战博弈、辅助决策、目标跟踪等应用要求低延时、快速闭环^[5-8]; (5)国产化要求: 为保证自主可控, 必须选择国产化处理器。

本文对比了目前常规处理器的特点以及它们在机载领域的应用现状, 分析了三类领域专用加速器在指令集、微结构、计算能力等方面的不同, 并针对机载数据关联应用进行了专用加速器设计和性能验证。

1 常规处理器

机载应用对于处理器有着复杂多样的性能要求, 如不同的计算精度、并行度、并行进程交互复杂度、计算吞吐量、实时性、能效比等, 目前市场上没有一种芯片能够满足所有不同类别的计算任务所需求的全部性能指标。针对各类机载计算任务, 既要保证运算快, 又要使得运算结果准确可靠或精度损失可接受。在机载领域中, 常规采用的处理器包括中央处理器(CPU: Central Processing Unit)、通用图形处理器(GPU: Graphics Processing Unit)、数字信号处理(DSP: Digital Signal Processing)、现场可编程门阵列(FPGA: Field Programmable Gate Array)等, 各常规处理器的特点比较如表 1 所示。

CPU 作为一种通用处理器, 在任何应用中发挥着不可替代的作用, 主要提供资源分配、调度管理, 以及部分运算能力。GPU 是一种通用的加速器件, 在面向大规模并行计算时能够发挥较高的性能, 但其底层硬件设计细节隐蔽, 使用时难以完全发挥其效能。DSP 和 FPGA 在传统信号处理中使用较多, 但基于 DSP 和 FPGA 技术生态较为封闭, 性能提升

表 1 常规处理器特点及存在的问题比较

类型	特点	存在的问题
CPU	通用处理器; 复杂的指令流水线、控制部件设计, 通用性强。	运算器面积占比低, 能效比低。核心数量有限。
GPU	通用加速器; 单指令多数据架构, 核心数量高, 可执行大规模并行计算任务。	硬件没有为特定算法设计专用通路, 能效比不如专用加速器。
FPGA	数据流处理, 计算资源可配置、灵活性高、高吞吐量, 适用于流式空间计算。	无浮点算力, 主频低, 开发周期长
DSP	哈佛结构, 浮点算力高, 信号处理能力强	技术生态封闭, 性能提升缓慢

缓慢, 难以实现资源共享, 因此除了在一些特定的应用场景, 目前 DSP+FPGA 的处理架构正在逐渐被 CPU+GPU 替代。为了追求更高的能效比, 在不考虑成本、通用性的情况下, 采用为机载应用的运算需求而定制的领域专用加速器不失为一种解决方案。

2 领域专用加速器

传统计算机体系结构的处理器重视通用性, 运算颗粒度小, 面对新兴的计算任务效率不高, 导致传统计算机体系结构处理器的性能提升放缓。边缘计算重视能效比, 出现了很多有别于传统计算机体系架构的处理器, 这些处理器针对某一领域算法共性特征定制微架构, 形成领域专用架构(DSA: Domain Specific Architecture)处理器, 从而实现能效比数量级提升^[9]。常见的例如面向深度学习领域加速设计的神经网络处理器(NPU: Neural network Processing Unit)、面向多媒体处理、高性能计算等应用的扩展指令集处理器、粗粒度可重构处理器(CGRA: Coarse-grained Reconfigurable Architecture)等。

2.1 智能计算加速器

智能计算算法主要包括神经网络运算、传统机器学习类运算等, 在智能计算发展的早期, 尚未出现专门针对智能计算领域研制的处理器, 通常选用现有的处理器去应对新生的人工智能。例如谷歌早期使用 1.6 万个 CPU 核进行猫脸识别的深度学习神经网络的训练; 英伟达使用 GPU 图形处理器用于神

神经网络训练;这些加速器虽然能够完成智能计算任务,尤其是GPU在训练端因其超高的算力仍在大规模采用,但由于这些加速器的通用性,难以在某些专用场景发挥高能效。

随着以深度学习为代表的人工智能的研究推广,各公司和研究机构研制了种类繁多的智能计算加速器^[10-11]。例如,谷歌专为机器学习应用TensorFlow打造了TPU处理器^[19],IBM公司针对脉冲神经网络SNN开发了TrueNorth处理器^[20],以及国内比特大陆、地平线等公司各自研发了AI处理器^[12,13]。这些处理器都是受人工智能、大数据、区块链、智能安防、自动驾驶等需求牵引,均衡运算精度、功耗、成本等因素而定制的、针对特定应用开发的领域专用加速器。其共同的特点是市场需求大、对若干种特定的应用加速能效比高。在机载环境中,这类加速器符合图像类应用的需求,但在空战博弈、辅助决策、航路规划等智能应用中,其低位宽算力为主的特点不能满足精度需求,需要进一步针对机载应用进行定制开发。

2.2 自定义指令集处理器

自定义指令集处理器包括完全自定义指令集和扩展指令集两种。完全自定义指令集处理器是摆脱原有的指令集架构(如ARM、X86等),从零开始设计的、面向特定应用的指令集处理器,又称专用指令集处理器(ASIP: Application-Specific Instruction-set Processor)。ASIP在设计时针对场景特定优化,所以在一些实际应用时可以表现出高能效、低面积、低成本。例如Synopsys公司推出的设计工具ASIP designer,用户可以采用nML语言描述自定义架构,通过工具生成一个完整的最小化的嵌入式处理器内核模型及开发环境^[14]。

扩展指令集是为了增强现有CPU在某一应用领域的处理能力而专门开发的指令,例如Intel的SSE、AVX系列扩展指令集,用于高性能计算和多媒体程序^[15];ARM的NEON、SVE扩展指令集,用于通用计算、高性能计算加速和多媒体处理^[16]。这些扩展指令集由官方设计,对应底层的汇编级运算加速,并提供可供用户调用的加速库函数。但由于受到商业因素限制,大部分CPU厂商不支持用户自定义扩展指令。

针对各家CPU指令集不开放的情况,加州大学

伯克利分校Krstic Asanovic团队设计了开源、指令集规范RISC-V^[17]。RISC-V包含基本指令集和扩展指令集,扩展指令集允许用户根据自己的需求进行自定义指令,具有灵活可扩展的特性。近年来基于RISC-V指令集规范开发的处理器在物联网、嵌入式、甚至高性能计算领域得到广泛应用,包括平头哥、Google、NVIDIA等公司都推出了基于RISC-V指令集的处理器内核^[18],欧洲启动处理器项目—EPI,用ARM作为主处理器,RISC-V内核做超算加速协处理^[19]。

面向特定应用设计的扩展指令集处理器可作为主处理器的协处理加速单元,结合Chiplet技术,设计成独立执行专用计算任务的芯片,实现算力扩展。进一步地,RISC-V专用加速单元以封装库函数的形式提供给国产化CPU端开发人员,不会破坏原有的自主可控生态,既能满足通用计算又可实现领域专用。

2.3 粗粒度可重构处理器

粗粒度可重构处理器(CGRA)是一种通过快速加载配置信息实现不同计算任务切换的处理器。CGRA将计算部分集成为较为规整的处理阵列(PE: Processing Element),PE每个周期从配置寄存器中加载配置信息,配置信息决定PE执行何种运算操作,在PE完成计算任务后,迅速对其加载新的配置比特流进行功能重构,由于单次配置信息量小,重构过程仅需几个到几百个时钟周期,因而可以达到快速可重构效果。

CGRA处理器的配置信息是由编译器生成的,核心步骤包括对应用程序进行代码分析,将应用程序划分为软件运行部分代码和硬件运行部分代码,通过代码变换及优化、任务划分、任务调度、映射配置等过程,最终生成CGRA控制器的控制码和数据通路的配置信息,这与传统的编译器有很大不同^[20]。实际应用中,保证高效易用同时又兼容现有的生态是CGRA处理器取得广泛应用的关键。

CGRA处理器目前已在视频解码、密码计算、通信基带处理等计算密集型领域得到应用,并且获得了较高的能效比^[21]。在机载领域应用中也可针对高密度异构算力的需求,开展芯片架构、算法及编译器研究,设计一种面向信号处理、智能计算算法的粗颗粒度可重构阵列处理器架构,提升机载信息处理算力。

3 机载处理平台选择

根据以上各种处理器的特点,结合各类机载应用的算法和数据流特性,在面向不同机载应用时可选择的处理平台如表 2 所示。

表 2 机载应用分类及处理平台选择

序号	机载应用	处理平台选择
1	通用计算 / 控制	主处理: CPU、 加速器: GPU/FPGA
2	图形 / 图像处理	主处理: CPU 加速器: GPU/NPU
3	信号处理	主处理: CPU 加速器: GPU/FPGA/CGRA
4	智能计算	主处理: CPU 加速器: GPU/NPU/FPGA

在通用计算 / 控制方面, CPU 的作用不可替代,但面对计算能力瓶颈,可通过 GPU、定制 FPGA 逻辑解决算法的中数据密集计算瓶颈,实现数据处理加速;在图形 / 图像处理方面, GPU 是最常使用的处理器,通过通用支持浮点运算的 GPU 或定制精度的 NPU 解决目标识别等机器学习算法瓶颈,实现图像处理加速;在信号处理方面,通过 GPU/FPGA/CGRA/ASIC 解决高带宽、通道型、密集矩阵、向量运算,实现信号处理专用加速;在智能计算方面,通过 GPU/NPU/FPGA 等完成行为树、决策树、遗传算法等完成符合机载数据精度要求的运算,实现智能计算加速。以上为面向机载应用的处理平台选择建议。下面以机载通用计算的一种应用(数据关联)为例,在 FPGA 平台上设计实现面向数据关联应用的专用加速器。

4 基于 FPGA 的数据关联专用加速器设计

在机载应用领域,传感器前端发送的目标数据与后端维护的数据进行关联是一个典型应用场景。随着传感器信息源种类增加,数据发送的规模不断增大,传感器前端目标数据按照一定的周期实时更新,对于后端数据处理的实时性要求不断提高。目前后端的数据关联算法主要在 CPU 上运行,处理平台不具备大规模数据的并行处理能力,面临着程序运行速度慢、算法性能受限的问题。本节对典型的数据关联算法进行了分析,并针对统计距离计算与分簇处理瓶颈在 FPGA 平台上进行了专用加速器设计。

4.1 算法流程

数据关联算法可以分为计算统计距离、对应关系分簇、最优关联、关联关系整合等步骤,如图 1 所示。其中计算统计距离部分的输入为目标数据集 A 和维护数据集 B。将目标数据集 A 和维护数据集 B 的坐标代入公式计算统计距离。这部分运算的特点在于计算密度高且不同坐标距离计算间无依存关系,较为适合使用通道加速器或多核并行计算方法来实现加速。

对应关系分簇处理时,先统计目标数据集元素坐标与维护数据集元素坐标的统计距离值是否小于阈值,只有小于阈值的元素才会进入下一步运算。再统计小于阈值的个数,若某个目标数据集元素坐标仅对应一个维护数据集元素坐标则代表这两个坐标为一对一(匹配)关系,即这两个坐标为同一目标的前后坐标,在得到该一对一关系后就将该索引关系存入最后输出的匹配数组中。对于某个目标数据集元素坐标与多个维护数据集元素坐标或多个目标数据集元素坐标与多个维护数据集元素坐标的对应,即一对多或多对多关系,则需要继续进行最优关联算法来判断这些坐标的匹配关系,此部分算法包含较多比较、判断的操作,比较适合 CPU 处理器来实现,因此,统计距离计算和分簇完成之后,将结果通过 PCIe 输出给 CPU 端继续完成最优关联和关联关系整合。

4.2 加速系统结构

面向数据关联算法的加速系统主要包括 CPU 和基于 FPGA 的专用加速器两部分,如图 2 所示。CPU 主要承载接收传感器目标数据、标准化格式列表、条件判断、数据粗处理以及将数据发送给 FPGA 等任务。而数据关联算法的统计距离计算与分簇处理等计算瓶颈则卸载到专用加速器进行,计算得到的结果送回 CPU 做进一步处理。主处理器与专用加速器之间通过 PCIe 总线互联。专用加速器数据存储分为两级,一是片上存储 BRAM,二是片外 DDR 存储,对于数据量较小的数据集,仅使用片上存储,对于数据量较大的数据集,优先使用片上存储,DDR 作为缓冲存储,分批次进行运算处理。

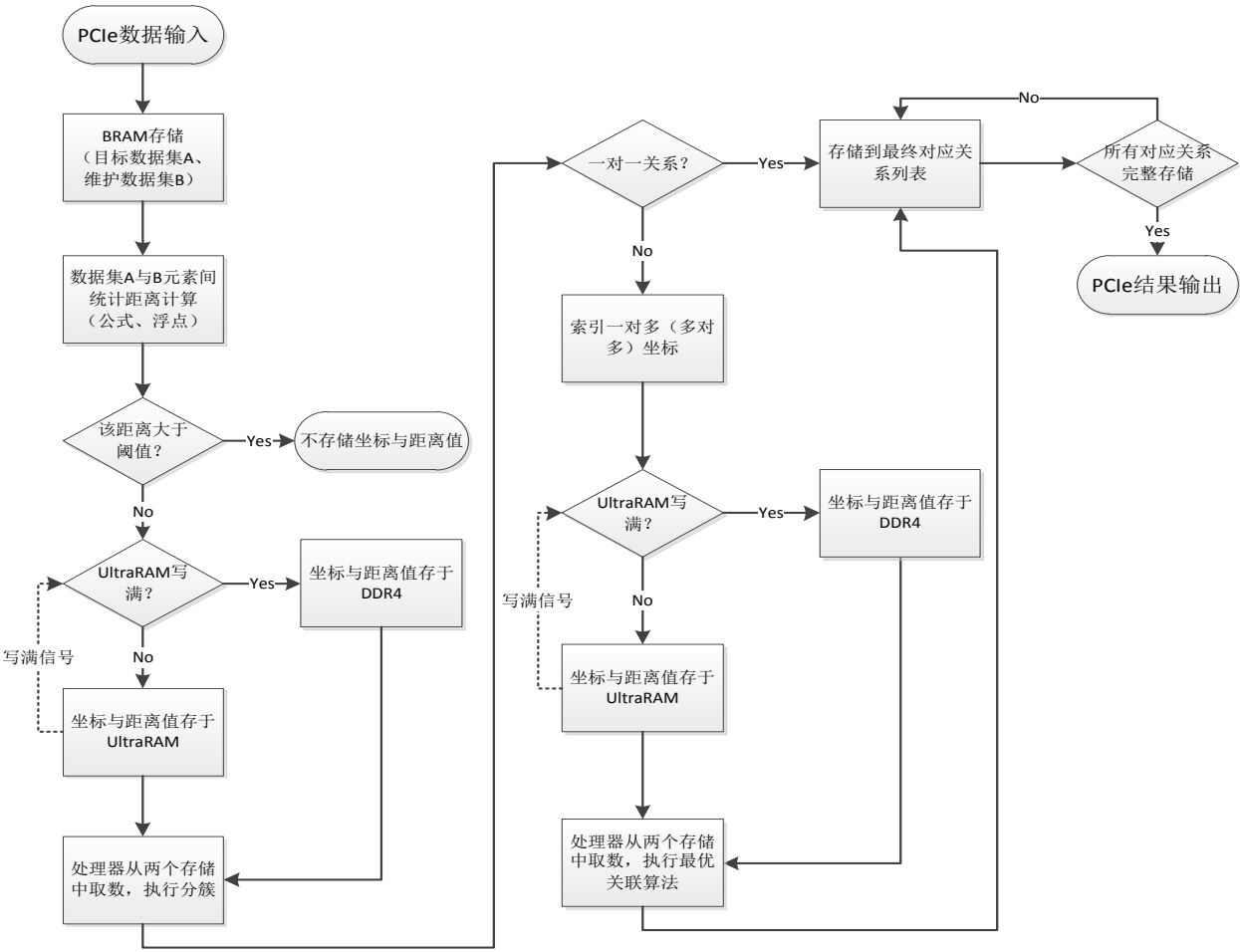


图 1 数据关联算法流程

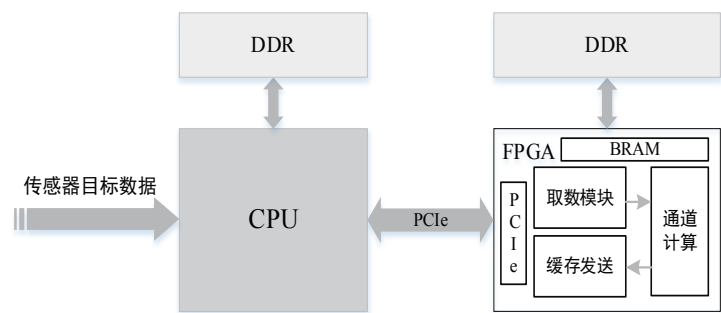


图 2 面向数据关联算法的加速系统

4.3 加速器实现

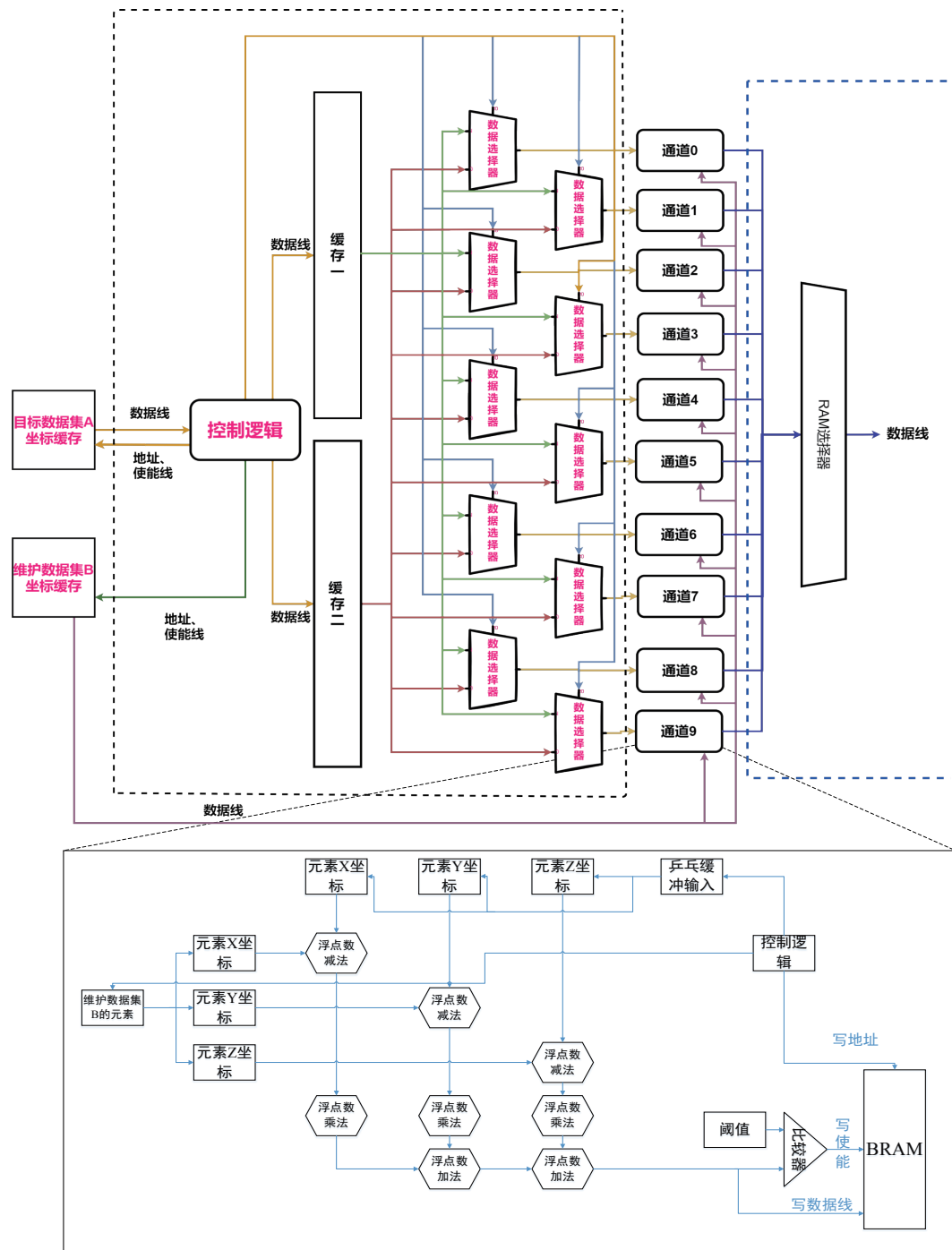


图3 面向数据关联算法的专用加速器架构

面向数据关联算法的专用加速器架构主要由三部分组成：一是用于从目标数据集A和维护数据集B中取数的取数模块，包括用于存储目标数据集A的乒乓缓冲和直接取维护数据集B的逻辑；二是用于计算的十个相同的通道模块，每个通道都由特定设计的浮点计算模块组成。浮点计算模块其包含了浮点数减法、浮点数乘法、浮点数加法、浮点数比

较单元，当一对来自目标数据集A、维护数据集B的元素坐标计算完后在浮点数比较单元与阈值进行比较，如果结果小于阈值，比较单元输出使能，将计算结果与这两个点的坐标保存到数据输出单元的数据缓存器中；三是缓存发送模块，通过数据选择器选中对应的缓存，将其结果从缓存中读出，并通过PCIe总线输出至CPU端。

加速器中设计有两个缓存, 当第一部分从乒乓缓存中取出的坐标计算完后直接切换到第二个缓存, 同时再往第一个缓存中写新坐标, 以此类推。通道加速器中各通道在计算时采用流水线处理方法, 每个时钟周期都有一个结果输出, 同时输入一对新的坐标开始计算, 计算的数据持续流入, 最终每对坐标的计算时间约为一个时钟周期。每个计算通道执行的向量浮点运算, 具体公式可根据应用的需求进行逻辑定制。分簇的步骤包括对现有关联矩阵的门限筛选比较(分簇步骤1), 以及对行列对应关系的筛选记录(分簇步骤2)两个步骤, 记录下过了门限的多对多子矩阵以及映射关系表。

4.4 测试结果

基于该技术搭建了数据关联算法专用加速系统, 如图4所示。系统主CPU为FT2000/4处理器, FPGA采用xilinx ZU7EG MPSoC, 系统输入为随机产生的1万组包含4000个元素的目标数据集、6000个元素的维护数据集。目前在实际应用中该算法完全运行在CPU上, 本文以FT2000/4 CPU完成数据接收并进行统计距离和分簇计算的耗时为基准, FT2000/4芯片集成4个飞腾自主研发的高性能处理器内核FTC663, 主频2.2 GHz, 采用其中一个内核运行算法, 测试结果如图5中CPU平均耗时所示。CPU完成数据接收后通过PCIE发送给FPGA端进行统计距离和分簇计算的耗时如图5中FPGA平均耗时所示。结果表明, FPGA对于统计距离的计算由在FT2000/4单核上的155 ms缩短至15.8 ms, 加速效果约为FT2000/4单核性能的10倍; FPGA对于分簇处理(分簇1+分簇2)的计算由在FT2000/4单核上的112 ms+65 ms缩短至2 ms+55 ms, 加速效果约为FT2000/4单核性能的3倍, 整体加速效果约为FT2000/4单核性能的5倍。值得注意的是, FPGA只是专用加速器的验证平台, 芯片运行主频只有200 MHz左右, 一旦形成专用ASIC, 芯片主频得到提升后整体加速性能可以获得更大提升。

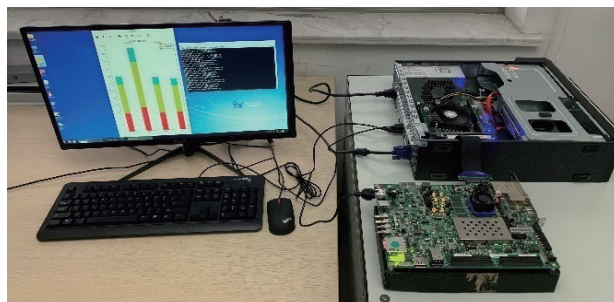


图4 FT2000/4+FPGA开发平台

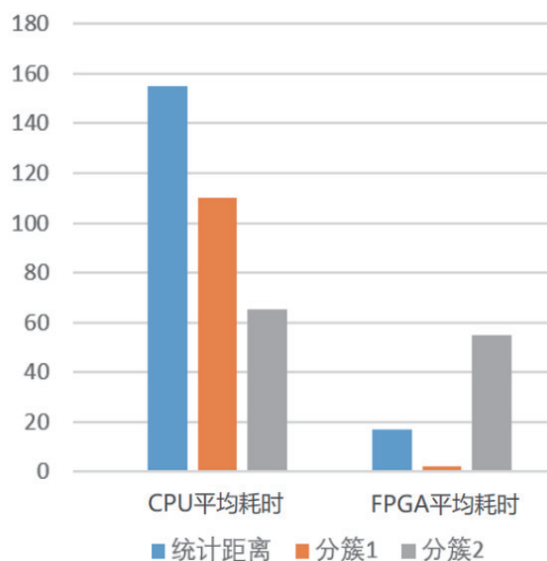


图5 CPU和FPGA平台的算法运行耗时统计

5 结论

在“CPU+X”异构计算系统中, CPU作为控制单元, 负责复杂的控制、调度等工作, “X”代表加速器, 负责大规模的并行计算或专业领域的计算任务, 除了通用的加速处理器外, 定制化的方案使得其数据精度和运算速度满足机载应用需求, 面向特定应用或者面向特定领域时可获得更高的能效比。本文针对机载领域数据关联应用进行了专用加速器设计, 测试结果表明该专用加速器对于统计距离计算、分簇处理等CPU平台的计算瓶颈有明显的加速效果, 验证了“CPU+专用加速器”是一种有效提升机载算力的技术路径。类似的, 在机载信号处理、智能计算领域也可采用定制化专用加速器实现机载计算系统的性能升级。

参 考 文 献

- [1] 杨伟. 关于未来战斗机发展的若干讨论[J]. 航空学报, 2020, 41(6): 524377-1-524377-12.
- [2] Wei P. Deep SAR imaging and motion compensation[J]. IEEE Transactions on Image Processing, 2021, 30(1):2232-2247.
- [3] Ming J, Liao G, Li J. Joint DOD and DOA estimation for bistatic MIMO radar[J]. Signal Processing, 2009, 89(2): 244-251.
- [4] 黄长强, 唐上钦. 从“阿法狗”到“阿法鹰”——论无人作战飞机智能自主空战技术[J]. 指挥与控制学报, 2016, 2(3): 261-264.
- [5] Tan K, Li W H. Imaging and parameter estimating for fast moving targets in airborne SAR[J]. Computational Imaging, 2017, 3(1):126-140.
- [6] Department of the Air Force. Department of Defense Fiscal Year (FY) 2021 budget estimates airforce justification book Volume 2 of 3 Research, development, test&evaluation [EB/OL]. (2020-02-12) [2020-06-02]. https://www.saffm.hq.af.mil/Portals/84/documents/FY21/RDTE_/FY21_Air_Force_Research_Development_Test_and_Evaluation_Vol_II.pdf?ver=2020-02-12-145218-377.
- [7] SCHUCKT. OODA loop 2.0 information, not agility, is life[J]. Alumni Magazine, 2017, 8(6):55-58.
- [8] 雷宏杰, 姚呈康. 面向军事应用的航空人工智能技术架构研究[J]. 导航定位与授时, 2020, 7(1): 1-11.
- [9] Takano S. Thinking Machines [M]. Applications, ASICs, and domain-specific architectures, 2021: 67-91.
- [10] Russell S J, Norvig P. 人工智能: 一种现代方法(第3版)[M]. 北京: 清华大学出版社, 2013:67-79.
- [11] Chen T, Du Z, Sun N, et al. DianNao: a small-footprint high-throughput accelerator for ubiquitous machine-learning[J]. AcmSigplan Notices, 2014, 49(4):269-284.
- [12] Salamin S, Zervakis G, Klemme F, et al. Impact of NCFET technology on eliminating the cooling cost and boosting the efficiency of google TPU[J]. IEEE Transactions on Computers, 2021, 3(12):1-1.
- [13] Akopyan F, Sawada J, Cassidy A, et al. TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neuro synaptic chip[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2015, 34(10):1537-1557.
- [14] Wheeler B. Bitmain SoC brings AI to the edge[J]. Microprocessor report, 2019, 33(2):25-27.
- [15] Halfhill T R. Horizon robotics eyes ADAS[J]. Microprocessor report, 2019, 33(4):29-32.
- [16] Eusse J F, Fernandez F, Leupers R, et al. Concurrent memory subsystem and application optimization for ASIP design[C]. International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation (SAMOS). IEEE, 2016:1-10.
- [17] Takahashi D. An implementation of parallel 2-D FFT using intel AVX instructions on multi-core processors[C]. International Conference on Algorithms & Architectures for Parallel Processing. 2012:197-205.
- [18] Stephens, Nigel, Biles, et al. The ARM scalable vector extension[J]. IEEE Micro, 2017(37): 26-39.
- [19] Lee Y, Waterman A, Avizienis R, et al. A 45nm 1.3GHz 16.7 double-precision GFLOPS/W RISC-V processor with vector accelerators[C]. ESSCIRC 2014 - 40th European Solid State Circuits Conference (ESSCIRC). IEEE, 2014:199-202.
- [20] Chen C, Xiang X, Liu C, et al. Xuantie-910: Innovating Cloud and Edge Computing by RISC-V[C]. IEEE Hot Chips 32 Symposium (HCS). IEEE, 2020:1-19.
- [21] 魏少军, 刘雷波, 尹首一. 可重构计算(第1版)[M]. 北京: 科学出版社, 2014:161-177.