基于 ARM + DLP + SRIO 的嵌入式智能计算系统研究

赵二虎,吴济文,查晶晶,郭 振,徐勇军
(中国科学院计算技术研究所专项技术研究中心,北京 100190)

摘 要: 以 x86 + GPU 为代表的当前主流 AI 计算平台,受限于功耗、体积、带宽、环境适应性等因素,无法适用于物端及边缘智能计算场景. 提出并研究了一种基于 ARM + DLP + SRIO 的嵌入式智能计算系统,从 AI 算力、能效比、IO 带宽三个方面分析了所提嵌入式智能计算系统的设计思路和技术优势,并实验验证了该系统的功能及性能指标. 实验结果表明:基于 ARM + DLP + SRIO 的嵌入式智能计算系统 AI 峰值算力达到 114.9TOPS,能效比达到 1.03TFLOPS/W,IO 带宽达到 20Gbps. 在智能计算系统领域,其能效比优于国内其它已知同类板卡或系统,嵌入式环境适应能力优于传统台式机和服务器,可作为物端及边缘环境下 AI 计算任务的通用硬件加速平台.

关键词: 人工智能; 深度学习处理器; 嵌入式智能计算系统; 串行 RapidIO; 能效比
中图分类号: TP389.1 **文献标识码:** A **文章编号:** 0372-2112 (2021)03-0443-11
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.12263/DZXB.20200415

Embedded AI Computing System Based on ARM + DLP + SRIO

ZHAO Er-hu, WU Ji-wen, ZHA Jing-jing, GUO Zhen, XU Yong-jun
(*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190, *China*)

Abstract: The existing artificial intelligent (AI) computing platform represented by x86 + GPU, limited by power consumption, dimension, bandwidth, environmental adaptability, and other factors, cannot be well adapted to the things and edge intelligent computing scenarios. We proposed an embedded AI computing system based on ARM (Advanced RISC Machine) + DLP (Deep Learning Processor) + SRIO (Serial RapidIO), and elaborated the design methods and technical advantages. In study, three aspects of the system were dissertated: AI computing performance, power efficiency, and IO bandwidth, and the function and performance of the system were verified by experiments. The results show that the peak performance of the embedded AI computing system based on ARM + DLP + SRIO is up to 114.9TOPS, the energy efficiency is up to 1.03TFLOPS/W, and the IO bandwidth is up to 20Gbps. In the field of AI computing systems, its energy efficiency is better than other similar boards or systems in China, and its embedded environmental adaptability is better than that of traditional desktops and servers, so it can provide a general hardware acceleration platform for AI computing tasks in things and edge computing scenarios.

Key words: artificial intelligent; deep learning processor; embedded AI computing system; serial RapidIO; power efficiency

1 引言

近年来,以深度学习为代表的人工智能技术快速发展,在语音识别、图像分类、自然语言处理、博弈决策、搜索推荐等应用领域取得了显著效果. 深度学习算法一般需要有强大的算力支持. 从算力部署位置来看,主要分为物端、边缘和云端. 云端计算环境多以追求算力为首要目标,对架构、体积、功耗等要求相对宽松,尤其在神经网络加速方面,云端已经形成以 x86 + GPU 为主流架构的智能计算生态. 然而,在物端及边缘计算场景中,受限于功耗、体积、带宽、环境适应性等约束条件,x86 + GPU 架构已“独木难支”,迫切需要研究更加适合

收稿日期:2020-05-06;修回日期:2020-08-03;责任编辑:覃怀银
基金项目:“十三五”领域基金(No.61403120111)

物端及边缘的智能计算系统.

物端计算[1~3]和边缘计算[2,4~6],虽然在学术概念上有所区分(例如,通信网络中将手机视为物端,将基站视为边缘;物联网中将摄像头视为物端,将视频网关视为边缘;航天系统中将飞行器视为物端,将发射塔站视为边缘),但在工程技术实现上却边界模糊,从计算机角度来看,两者最大的共同点便是均属于嵌入式系统.综合考虑物端及边缘智能计算需求及其嵌入式应用特征,本文提出并研究了一种基于 ARM + DLP + SRIO 的嵌入式智能计算系统,旨在为物端及边缘环境下的 AI 计算任务提供通用硬件加速平台.首先将对嵌入式智能计算系统面临的问题进行描述;然后对嵌入式智能计算系统的硬件架构进行设计;接着将在硬件架构的基础上搭建嵌入式智能计算原型系统,详细阐述系统各部分工作机理;最后对嵌入式智能计算系统进行性能评测,并给出结论.

2 问题描述

2.1 AI 算力及能效比问题

在嵌入式计算环境中实现智能加速,首要任务是解决嵌入式系统的 AI 算力问题,而所有的 AI 算力又必须在低功耗条件下执行.以 ARM 为代表的通用嵌入式处理器,从内部结构上来看,70% 晶体管都是用来构建 Cache(高速缓冲存储器)和一部分控制单元,负责逻辑运算的算术逻辑单元 ALU(Arithmetic and Logic Unit)模块并不多.控制单元等模块的存在都是为了保证指令能够一条接一条的有序执行.这种通用性结构对于传统的指令密集型计算模式非常适合,但对于深度学习这种数据密集型计算则“力不从心”.如文献[9]所述,在高通 Snapdragon 820 ARM 处理器上运行 1.0 MobileNet-224[10]神经网络模型,实测处理一张 720 × 1280 的图片需要 1879.2ms,倘若处理一张 1280 × 1280 的图片,则至少需要 3340.8ms,计算方法如式(1).假设光学传感器以 10fps 的帧率产生分辨率为 1280 × 1280 的图像,如果要对每帧图像进行实时目标检测,单一的 ARM 嵌入式处理器完全无法胜任上述计算任务,其实际算力与需求算力相差约 17.78 倍,如式(2).

$$\frac{1280\times1280}{720\times1280}\times1879.2\approx3340.8\text{ms} \tag{1}$$

$$3340.8\times10\div1879.2\approx17.78 \tag{2}$$

以 GPU 为代表的图形处理器,因其强大的并行计算能力,适用于神经网络中大量的浮点计算以及矩阵和向量运算,成为目前使用最多的智能加速器.但是 GPU 为了保持既有的图形处理能力,内部用来运行 AI 计算任务的逻辑单元大概占芯片面积的 40% 左右,多数的晶体管并不会用来处理 AI 计算任务,导致 GPU 的功耗一直居高不下.目前 GPU 生态布局主要在云端,典型产品如 NVIDIA Tesla V100,功耗高达 300W;物端只推出了少数几款示范应用,比如 NVIDIA 针对自动驾驶领域推出的 Drive PX 平台,虽然算力达到 20TFOPS,但功耗仍高达 80W,且必须使用主动散热方式,否则难以适应严苛的使用环境.因此在嵌入式智能计算场景中,GPU 并不被业界所看好,也未被广泛采用.

正是高性能、低功耗、强实时性的苛刻要求驱使嵌入式智能计算系统必须采用异构多处理器架构,此处所述的异构多处理器与用于科学计算的对称多处理器是完全不同的.因此如何设计一种通用嵌入式处理器与专用智能处理器相互协同的混合异构计算平台,拥有出色的性能功耗比,成为嵌入式智能计算系统面临的首要问题.

2.2 IO 互连问题

人工神经网络是一种数据密集型计算模型,不仅体现在神经网络的多层结构和多参数特征上,而且体现在数据源的多模态和连续性方面.多模态主要强调数据源的数据类型多,数据格式各不相同,例如可见光、红外、超声波、毫米波等;连续性则强调数据流的源源不断,且存在一定的带宽差异性,如传感器一以 20fps 的帧率产生 5120 × 5120 的图像、传感器二以 20fps 的帧率产生 1280 × 1280 的图像、传感器三以 10fps 的帧率产生 2048 × 2048 的图像.将上述三类传感器所产生的数据流合并统计,均按照 16bit 量化,带宽需求约为 9.6Gbps,计算方法如式(3).而目前嵌入式系统中最常用的高速接口为千兆网口(传输速度 1000Mbps),如果采用 TCP 协议,其数据打包效率约为 79%,则千兆网口的有效数据带宽约为 790Mbps,与需求带宽相差约 12 倍.

$$5120\times5120\times20\times16+1280\times1280\times20\times16+2048\times2048\times10\times16\approx9.6\text{Gbps} \tag{3}$$

如果单从带宽需求来看,PCIe[11]作为桌面计算机和服务器中广泛使用的高速串行总线,是能够满足 10Gbps 以上量级数据传输需求的.例如采用 PCIe3.0 标准[12],单 lane 的传输速率达到 8GT/s,去除编码开销(编码效率为 128b/130b)后,单 lane 有效数据传输速率为 7.8768Gbps,两个 lane 即可满足式(3)带宽需求.然而,PCIe 并不适合作为嵌入式智能计算系统的高速互联总线,原因有四:一是 PCIe 本为 Intel x86 体系下产物[11],主流嵌入式处理器(如 ARM、PowerPC)对 PCIe 支持程度并不太好;二是 PCIe 属于树形拓扑结构[11],任何 PCIe 通信链路的建立必须有 RC(Root Complex,根联合体)和 EP(End Point,终端设备),而智能加速芯片大多为协处理器,可视为 EP 设备,很难依靠 PCIe 建立起多个 EP 设备之间的协作网络;三是 PCIe 协议缺乏链路

控制报文[15],在误码条件下会弃包来防止链路堵塞,且误码条件机制不可配置,当链路需要询链时数据包通常会被丢弃,不适用于对可靠性要求较高的特殊应用领域;四是 PCIe 不支持数据的分段处理[11],即总线上正在处理的事务会在整个事件段内占用总线,阻塞了其它事务对总线的使用,因此不适用于多模态感知数据嵌入式互联场景.

因此,如何构建一套高速、高可靠、可灵活扩展的嵌入式 IO 互联总线,成为制约嵌入式智能计算系统能否充分发挥其智能算力的关键问题.

3　硬件架构

针对上述两个主要问题,本文提出并研究了基于 ARM + DLP + SRIO 的嵌入式智能计算系统,系统架构如图 1 所示,主要包括深度学习处理器 DLP 子系统、嵌入式处理器 ARM 子系统、IO 互连子系统.

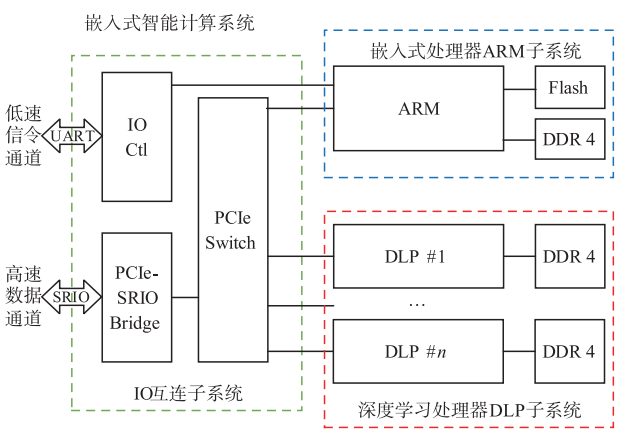


图1　基于ARM+DLP+SRIO的嵌入式智能计算系统架构图[21]

其中,针对 AI 算力及能效比问题,本文设计了 ARM + DLP 的嵌入式异构智能计算架构. 该架构中,DLP 主要负责神经网络模型中各类算子的运算加速(即数据密集型计算任务). 例如,卷积神经网络 VGG19 中有三个典型层,分别是卷积层、池化层以及全连接层. 相应地,这三种层涉及三种常见算子——全连接、卷积和池化,而这三种算子都是向量或者矩阵(高维向量)的操作,即可向量化操作[7],最适合交给 DLP 来加速运算. ARM 主要负责智能计算任务的数据传输、流程管理和 DLP 驱动管理(即指令密集型计算任务). 例如,从可见光传感器获取的原始图像数据需要先经过 ARM 的内存,然后再转发给 DLP 的显存;DLP 的智能计算结果需要先发给 ARM,然后再转发给其它计算单元;流程管理涉及到数据的输入、缓存、输出、模型加载、模型切换等;DLP 驱动管理涉及到系统设备加载、数据并行、模型并行、显存管理、多核分配等. ARM 与 DLP 之间通过 PCIe 总线进行互连. 采用 ARM + DLP 异构智能计算架构,不仅弥补了 ARM 处理器无法快速处理大数据量神经网络计算的短板,而且在性能功耗比方面较 x86 + GPU 架构有显著提升.

针对 IO 互连问题,本文采用了 SRIO[13,14]作为嵌入式智能计算系统的外部总线. SRIO 体系结构是为了满足高性能嵌入式系统对数据交换的大带宽、低时延、高灵活性和高可靠性等要求而产生和发展的[16]. 在带宽上,SRIO2.0 的单通道信号速率可达 6.25GHz,双通道并行即可达到 12.5Gbps 的通信速率,满足式(2)带宽需求;在可靠性上,SRIO 在物理层定义了纠错及重传机制,并且在消息事务层采用了系统级流量控制策略和错误管理机制,可实现错误的快速诊断和自我恢复;在可扩展性上,SRIO 的扁平化拓扑结构可由任意个终端器件和交换器件组成,终端器件间采用对等架构,不采用主从架构,克服了 PCIe 树形拓扑结构的弊端;在兼容性上,SRIO 目的是提出一种开放的嵌入式互连标准,它由一批嵌入式系统和半导体的研究机构和领导厂商开发,并由 RapidIO 行业协会监管维护,可以确保该技术最大限度满足嵌入式应用而非某些公司的利益需求. 因此,嵌入式智能计算系统优选 SRIO 作为其外部 IO 互连总线.

3.1　深度学习处理器 DLP 子系统

深度学习处理器 DLP 子系统硬件架构如图 2 所示. 本系统中的深度学习处理器选择 MLU100[8]. MLU100 是寒武纪推出的面向边缘推理的深度学习专用加速芯片,该芯片基于寒武纪 MLU(Machine Learning Unit)v01 架构,采用自主知识产权的 Cambricon-ISA 指令集,支持视觉、语音、自然语言处理、传统机器学习等多模态人工智能应用,与 GPU 处理器相比有更优的能效比. MLU100 采用多核处理器架构,各个核之间通过片上网络(Network on Chip,NoC)与 4 个 DDR 控制器相连,每个 DDR4 控制器支持 64 位内存数据宽度. 本系统的 4 个 DDR 通道共构成 256 位 8GB 的内存空间. 此外,

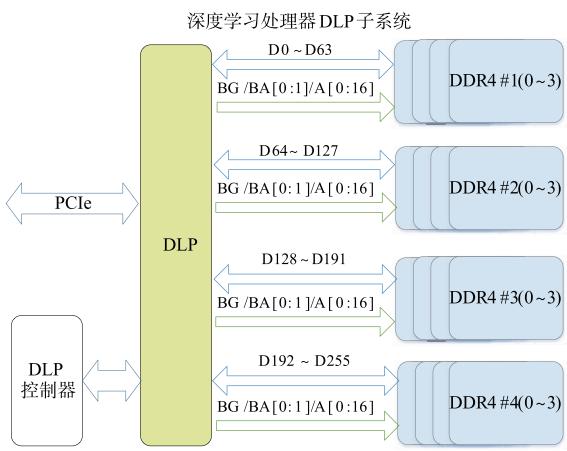


图2　DLP子系统硬件结构[21]

DLP 子系统中配备 1 片 DLP 控制器,负责 MLU100 的工作参数管理和工作状态监控.

MLU100 的软件开发环境为 Cambricon NeuWare[17,18],如图 3 所示,支持各类主流编程框架(如 Caffe、TensorFlow、Pytorch、MXNet 等). 用户可面向上述编程框架,便捷地在 MLU100 上开发和部署深度学习应用. Cambricon NeuWare 提供了完整的运行时系统和驱动软件,方便系统快速集成. Cambricon NeuWare 还提供了包括应用开发、功能调试、性能调优等在内的一系列工具,其中应用开发工具包括机器学习库、运行时库、编译器、模型重训练工具和特定领域(如视频分析领域)SDK 等;功能调试工具可以满足编程框架、函数库等不同层次的调试需求;性能调优工具包括性能剖析工具和系统监控工具等. 上层的机器学习应用可以直接采用各种编程框架的编程接口,间接通过 Cambricon NeuWare 机器学习库(Cambricon Neuware Machine Learning Library, CNML)调用 Cambricon NeuWare 运行时库(Cambricon Neuware Runtime Library, CNRT)进行软件编程;也可以直接调用 CNRT,运行上述过程所生成的离线模型,减少软件架构的中间开销,提高实际运行效率.

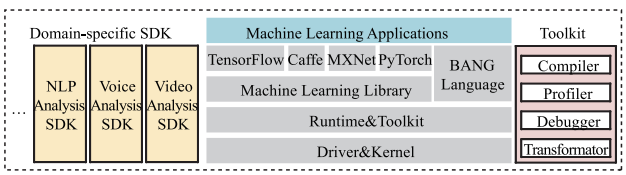


图3 DLP软件开发环境

3.2 嵌入式处理器 ARM 子系统

嵌入式处理器 ARM 子系统主要由 ARM 处理器、千兆网口、DDR4 SDRAM、SD 卡或 eMMC 存储器、NAND Flash、RS232 串口、EEPROM 和温度传感器、NOR Flash、电源网络、时钟网络、复位网络、PCIe 链路等组成,硬件架构如图 4 所示[21]. ARM 处理器的功能定位主要包括智能计算任务的数据传输、流程管理和 DLP 驱动管理,因此在 ARM 选型过程中主要从以下三个方面来考量:一是能否驱动 DLP 深度学习处理器;二是能否满足嵌入式智能计算系统对低功耗及高速 IO 通信的需求;三是能否支持多业务多流水线并行优化. 通过参考 MLU100 的官方资料[8],并综合考虑内核架构、指令集、主频、SDRAM 控制器、PCIe 控制器、PCIe BAR 空间等因素,本系统选定了 ARMv8 作为嵌入式智能计算系统的主处理器内核架构,同时集成了 4 个 ARMv8 架构内核以及报文处理、安全管理、IO 控制器等 IP 核. ARM 上电后首先完成对操作系统的启动及外部设备的初始化和挂载,之后启动用户进程,并行同步开展多业务数据流的接收、缓存、预处理、拷贝、DLP 调度、结果后处理等.

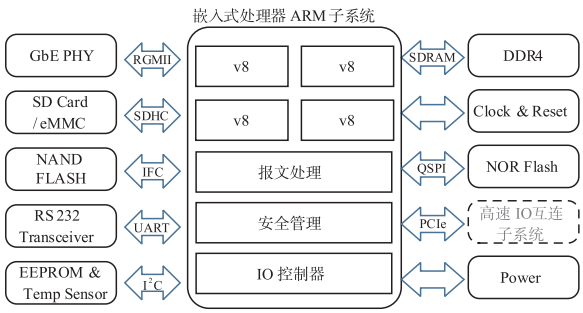


图4 ARM子系统硬件结构

ARM 嵌入式系统的通用软件栈[19]如图 5 左半部分所示. 硬件抽象层提供对设备及其他硬件原语的统一抽象,把软件从具体的设备和处理器的内部结构中抽象出来. 操作系统层负责控制基本的系统资源,如进程调度和内存管理. 进程间通信层提供抽象的进程通信服务,例如该层可以提供 PCIe 数据接收进程与 SDRAM 访问进程之间的数据接口,而无需关心两个进程分别运行在哪个处理器内核之上. 应用专用库负责提供面向特定应用所需的计算和通信功能. 顶层应用程序利用上述层来提供终端服务功能. 然而,当 ARM 子系统作为嵌入式智能计算系统的主控单元时,主要服务于异构智能计算任务,需支持 DLP 的深度学习编程框架,因此 ARM 嵌入式系统软件栈将面向智能计算任务进行优化,如图 5 右半部分所示[18].

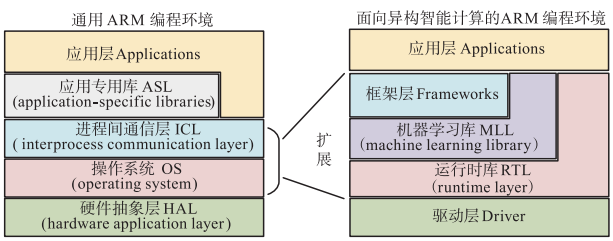


图5 面向智能计算的ARM软件栈优化

3.3 IO 互连子系统

IO 互连子系统负责在 ARM 子系统、DLP 子系统、外部其它系统(如传感器系统、机电系统、决策系统、存储系统等)之间搭建数据互连通道,包括低速信令通道和高速数据通道. 其硬件组成如图 6 所示. 低速信令通道主要实现 ARM 子系统与外部系统之间的信令交互,如控制指令、心跳指令、复位指令、休眠/唤醒指令等,通信接口采用 UART 串口. 高速数据通道完成嵌入式智能计算系统与外部系统之间的大数据量、高带宽、强实时性数据的交互,对外通信接口采用 SRIO.

高速数据通道主要由 PCIe Switch 模块和 PCIe-SRIO 桥接模块组成. 通过 PCIe Switch 模块的级联,搭建了 PCIe 互联拓扑树,实现多个 DLP 深度学习处理器的并行接入以及 PCIe-SRIO 桥接模块的接入. 在 PCIe

总线中,PCIe Switch 由 1 个上游端口和若干下游端口组成. PCIe Switch 的上游端口与 ARM 直接相连,下游端口与 PCIe 终端设备相连,或者与下一级 PCIe Switch 的上游端口相连. 相对于 PCIe 互联拓扑树来说,DLP 处理器、PCIe-SRIO 桥接模块都属于 PCIe 终端设备. 连接到 PCIe 互联拓扑树上的所有设备均可以实现端到端相互通信,因此接入到 PCIe 拓扑树中的深度学习处理器,其智能计算资源可以被其它端口设备所共享. 嵌入式智能计算系统对外采用 SRIO 作为高速数据通道. 虽然 SRIO 和 PCIe 都具备高速数据通信能力,但是这两种总线技术有着不同的协议,如要实现互连,需要通过 PCIe-SRIO 桥接模块[20,22,23]在两者之间传递事务.

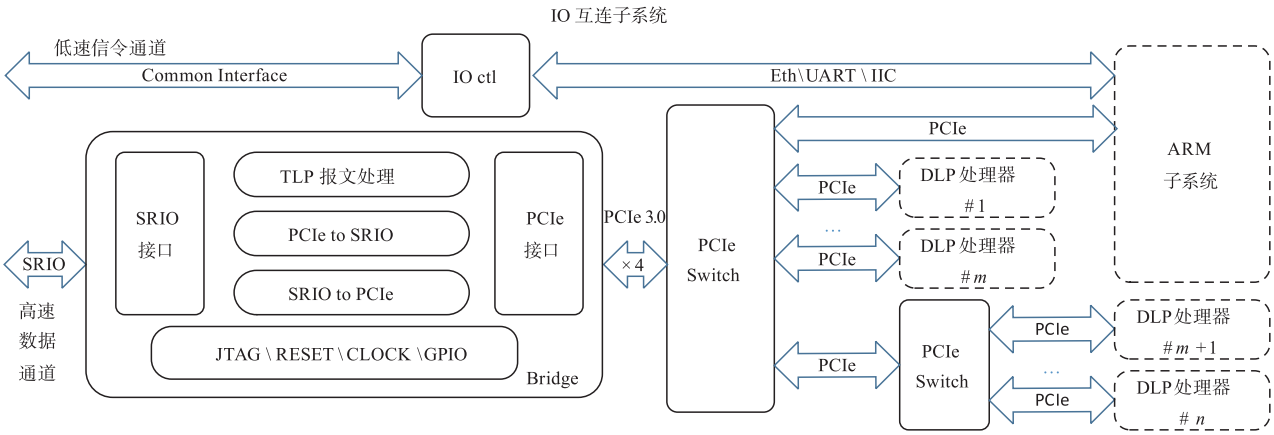


图6　IO互连子系统硬件组成框图[21]

4　原型系统

基于上述研究思路,作者设计并研制出了基于 ARM + DLP + SRIO 的嵌入式智能计算系统原理样机. 板卡采用 Open VPX 3U 结构,依靠金属结构件导冷散热. VPX 连接器上拥有 SRIO、UART 通信接口,同时由 VPX 背板提供电源. VPX 结构为嵌入式智能计算系统提供了灵活便捷的系统扩展能力,且具有良好的环境适应性. 软件上运行 Ubuntu16.04 操作系统. ARM 嵌入式系统编程,需采用交叉编译的方式生成目标板上可执行程序,本系统采用 ARM GCC 交叉编译工具,包含交叉编译器、连接器及目标库.

4.1　硬件启动

嵌入式智能计算系统的电源网络异常复杂,压轨多达 10 种,必须精确控制各路压轨的上电时序,才能保证系统的稳定启动. 设计过程中,本系统对各路压轨的输出使能进行控制,且每种电压输出电路具有 PG(Power-Good)开漏输出指示信号. 由于开漏输出电路具有“与(&)”操作特点,因此本系统利用该特点,将上级各路电压网络的 PG 信号进行“&”操作,待上级电压网络均正常后,方可产生下一级电压网络输出使能信号.

板级复位网络为嵌入式智能计算系统的各个子系统提供复位信号,具有上电复位和软复位功能. 上电后,本系统持续对末级电压网络的 PG 信号进行检测,待检测到 PG 信号有效后,输出全局复位信号,低电平有效. 其中,末级电压网络的 PG 信号由 PG_1.2V_DDR、PG_0.8V_CORE 开漏输出电路“&”产生. 综上所述,嵌入式智能计算系统的上电时序与复位流程如图 7 所示.

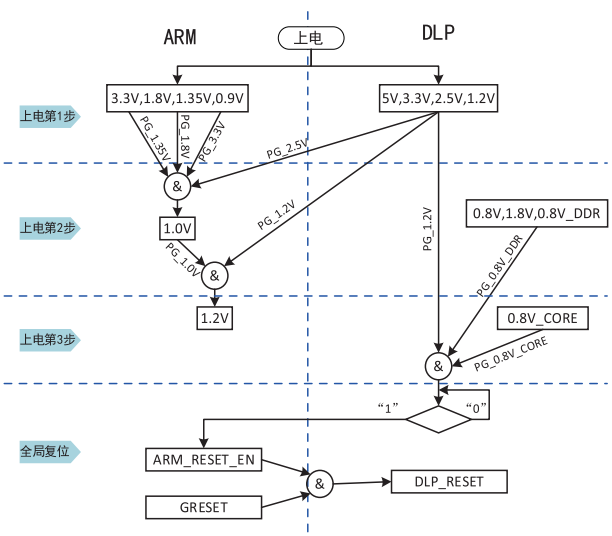


图7　嵌入式智能计算系统的上电时序与复位流程

4.2　软件流程

嵌入式智能计算系统的软件设计主要面向智能计算业务,包括高速数据传输模块、低速指令收发模块、任务调度模块、推理前数据缓存模块、推理计算模块以及推理后结果处理模块. 其中,高速数据传输模块负责完成 SRIO 高速数据传输与缓存处理;指令收发模块主要通过 UART 接口完成用户指令的接收、处理和反馈. 任务调度模块是嵌入式智能计算系统执行推理计算任务的指挥中枢,负责完成多种业务数据与多个网络模型的管理与调度. 以可见光图像目标检测和红外图像

目标检测这两个并发智能计算任务为例,任务调度模块首先根据 UART 传输的指令完成两个 AI 算法模型的加载;然后对 SRIO 传输的可见光图像数据和红外图像数据进行分区实时缓存,并按数据类型分发至下一级推理数据缓存模块.

推理前数据缓存模块主要功能是为 AI 算法模型的运行分配 CPU 内存空间与 DLP 内存空间资源,并建立二者内存拷贝传输描述符映射. 为提高内存使用效率,CPU、DLP 内存资源分配采取按需分配原则,具体将依据不同 AI 算法模型对源数据的批处理能力来定,以图像目标检测为例,其批处理能力主要表现为图像数量(N)、通道数(C)、图像高度(H)、图像宽度(W)等参数. 为提高推理计算效率,推理数据缓存模块将图像数据分批发送至下一级推理计算模块,每批图像数量由实际算法模型参数——图像数量 N 来决定.

推理计算模块负责完成 AI 算法模型文件加载、模型输入输出参数提取、DLP 推理计算资源调用以及 CPU 与 DLP 计算同步,并处理上一级推理前数据缓存模块发送过来的图像数据,同样会涉及到数据缓存操作. 如 2.1 节所述,嵌入式智能计算系统需要解决多业务多模态数据的智能计算问题. 为提高智能计算效率,推理计算模块需要完成多业务并发的 AI 计算任务. 在推理执行过程中,ARM 对 DLP 进行资源调度,将推理过程进一步拆分为多个神经网络算子,并交由 DLP 多核并行运算. 推理后结果处理模块负责处理每批图像推理计算后生成的结果数据,主要包括结果可视化、结果数据封装、结果报文发送等.

上述每个模块通过创建独立线程来实现,线程间通信采用同步阻塞队列方式,智能计算软件的工作流程如图 8 所示.

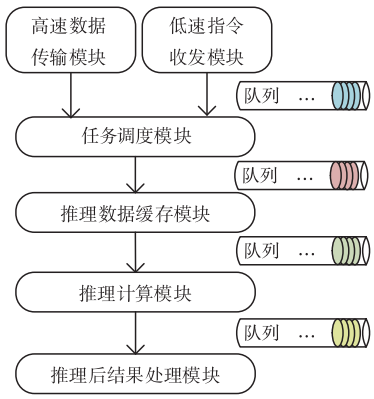


图8 主控软件工作流程图

5 系统评测

本节将主要从 AI 算力、能效比、IO 通信能力等三个方面对嵌入式智能计算系统进行评测. 这三个方面正是物端及边缘智能计算所关注的核心性能参数.

5.1 实验平台

搭建了一套完整的嵌入式智能计算系统,整机采用 VPX 3U 结构,含有 1 块标准 3U VPX 嵌入式智能计算模件、1 块 VPX 背板、1 台笔记本调试终端、2 台直流稳压电源,如图 9 所示.

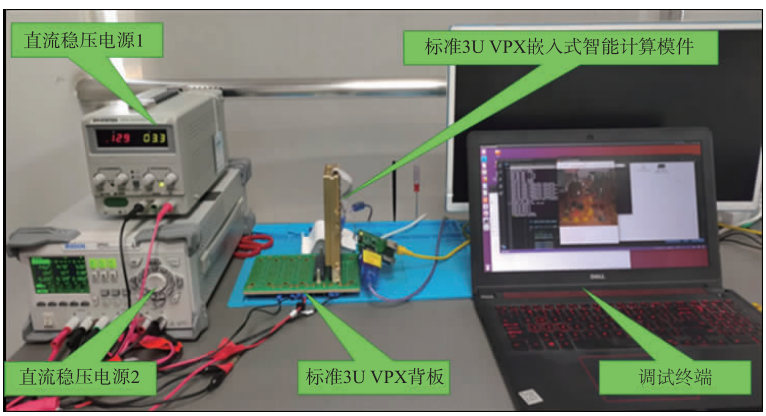


图9 嵌入式智能计算系统工作环境

其中,直流稳压电源通过 VPX 背板直接提供电源给 3U VPX 嵌入式智能计算模件. 笔记本电脑作为调试终端,通过千兆网口与智能计算模件相连,负责发送原始图像数据;智能计算模件接收到原始图像数据后,执行 AI 网络模型的推理计算,之后将计算结果反馈给调试终端;调试终端接收到推理计算的结果后,进行算力统计,并显示目标检出后的图像.

5.2 AI 峰值算力实验

为了测试嵌入式智能计算系统的峰值算力,首先设计了一个仅含有整形 INT8 乘法算子的测试矩阵(代表一幅二维图像),并且可通过参数设定来调整矩阵乘的运算规模,其次设定矩阵乘的循环次数,然后通过库函数读取系统每次执行矩阵乘的运算时间,最后求取平均值,获得系统的峰值算力. 测试程序运行流程

如下:

步骤 1　设置两个输入矩阵的运算规模,参数为 $N=1, C=16, H=1, W=2048$. 其中 N 表示每个批次计算的矩阵数量(也可视为一次批处理的图像数量),C 表示通道数(例如黑白图像的通道数 $C=1$,而 RGB 彩色图像的通道数 $C=3$),H 表示矩阵在竖直方向的元素个数,W 表示矩阵在水平方向的元素个数. 每个矩阵元素值随机生成. 所设定的矩阵如图 10 所示.

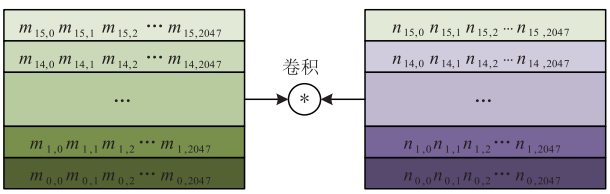


图10　峰值算力测试的两个输入矩阵设定

步骤 2　先后在 ARM 和 DLP 上申请预设值的内存空间,将数据从 ARM 拷贝到 DLP. ARM 的内存空间设定为 2GB,DLP 的内存空间设定为 16GB.

步骤 3　在 DLP 上循环执行矩阵运算,将运算结果拷贝回 ARM,释放 ARM 和 DLP 空间. 运行“./peak-dp 32-ln 4000-eln 20”命令,其中 dp = 32 表示数据并行度,ln = 4000 表示循环次数,eln = 20 表示循环体内的扩展循环次数. 总的运算量用 ops(operations)表示,如等式(4)所示:

$$\begin{aligned}\text{ops} &= 2W \times W \times \ln \times \text{eln} \times \text{dp} \times C \times N \times H \\ &= 2 \times 2048 \times 2048 \times 4000 \times 20 \times 32 \times 16 \\ &= 343.597 \times 10^{12}\end{aligned} \tag{4}$$

步骤 4　调用 NeuWare 库函数,统计在 DLP 上完成相应矩阵计算所消耗的时间,计算峰值算力 OPS(Operations Per Second). 假设总的运算量为 ops,运算时间为 t,则峰值算力可表示为:

$$\text{OPS} = \text{ops}/t \tag{5}$$

执行 10 次测试,测试结果如表 1 所示. 表中运算时间指 DLP 内部推理时间,不包括数据拷贝时间. 可以看出,嵌入式智能计算系统的平均峰值算力达到 114.9×10^{12}OPS,即 114.9TOPS(Tera Operations Per Second). 值得说明的是,因受限于测试数据、算子类型、硬件配置、测试方法等因素,峰值算力测试结果仅代表特定测试环境下的计算性能,不能代表本系统的最高性能,因此表中测试结果只作为本系统 AI 算力的一个参考值.

表 1　峰值算力评测结果

测试次序	运算量/ops	运算时间/ms	峰值算力/OPS
1	343.597×10^{12}	2990.140110	114.91×10^{12}
2		2990.140110	114.91×10^{12}
3		2990.140110	114.91×10^{12}
4		2990.166131	114.909×10^{12}
5		2990.114088	114.911×10^{12}
6		2990.166131	114.909×10^{12}
7		2990.114088	114.911×10^{12}
8		2990.270223	114.905×10^{12}
9		2990.270223	114.905×10^{12}
10		2990.114088	114.911×10^{12}
平均值			114.9091×10^{12}

5.3　能效比实验

在嵌入式智能计算系统中,以光学图像作为待测原始数据,移植并运行实际的目标分类网络模型,分别测试该系统对不同网络模型的浮点算力 TFLOPS(Tera FLoating-point Operations Per Second),然后根据模型运行时的实时功耗可以推算出能效比 EER(Energy Efficiency Ratio),单位是 TFLOPS/W. 计算方法如下[25]:我们用乘-加运算次数 MAdd(Multiply-Adds)和浮点运算次数 Gflops(Giga floating-point operations)来衡量模型的实际运算量. 假设采用滑动窗实现卷积且忽略非线性计算开销,则卷积核 ck(convolutional kernel)的乘-加运算次数为:

$$\text{MAdd}_{\text{ck}} = 2HW(C_{\text{in}}K^2 + 1)C_{\text{out}} \tag{6}$$

其中,H、W 和 C_{in} 分别为输入特征图的高度、宽度和通道数,K 为卷积核宽度(假设是对称结构),C_{out} 为输出通道数. 全连接层 fcl(full connected layer)的乘-加运算次数为:

$$\text{MAdd}_{\text{fcl}} = (2I - 1)O \tag{7}$$

其中,I 为输入维数,O 为输出维数. 在嵌入式智能计算系统中,一次乘-加运算包含 8 次浮点数操作,那么单个模型的浮点运算量为:

$$\begin{aligned}\text{Gflops} &= 8 \times \text{MAdd} \times 10^{-9} \\ &= 8 \times \sum(\text{MAdd}_{\text{ck}} + \text{MAdd}_{\text{fcl}}) \times 10^{-9}\end{aligned} \tag{8}$$

如果能够实验统计出目标分类的帧率 fps,就可以推导出相应的 AI 算力需求,计算式如下:

$$\text{TFLOPS} = \text{Gflops} \times \text{fps} \times 10^{-3} \tag{9}$$

假设当前的系统功耗为 P,那么能效比为:

$$\text{EER} = \frac{\text{TFLOPS}}{P} = \frac{\text{Gflops} \times \text{fps} \times 10^{-3}}{P} \tag{10}$$

对常用目标检测/目标分类网络进行嵌入式移植和评测. 测试前,记录测试图片大小、图片批处理数量、单模型运算量,并统计各个模型的算力需求,测试过程中记录实际运行功耗(单位:W),计算出相应的能效比,结果如表 2 所示. 嵌入式智能计算系统的平均能效比达到了 1.03TFLOPS/W.

表 2 基于嵌入式智能计算系统的目标检测/分类模型评测结果

模型	图片大小	图片数量	乘-加运算量 MAdd($\times 10^9$)	浮点运算量 Gflops	帧率 fps	算力需求 TFLOPS	功耗 W	能效比 TFLOPS/W
Yolov3[27]	416 × 416	1000	20.29	162.32	124.123	20.14765	48	0.419743
Yolov3	1024 × 1024	500	20.29	162.32	118.199	19.18606	34	0.564296
Yolov3	1280 × 1280	500	20.29	162.32	125.9	20.43609	22	0.928913
VGG-19[26]	224 × 224	1000	19.6	156.8	283.097	44.38961	46	0.964992
Resnet-34[24]	224 × 224	1000	3.6	28.8	1529.88	44.06054	18	2.447808
Resnet-50[24]	224 × 224	1000	3.8	30.4	820.366	24.93913	23	1.08431
Resnet-152[24]	224 × 224	1000	11.3	90.4	333.122	30.11423	37	0.813898
平均值								1.031994

本文分析了国内外业界领先的 15 种 AI 加速硬件平台,从官方资料、业界评测、实验室实测、理论推算等不同途径获取不同 AI 加速硬件平台的半精度浮点数 FP16 算力及相应的功耗,并依据等式(10)计算其能效比,结果如表 3 和图 11 所示.

表 3 不同 AI 加速硬件平台的能效比情况

型号	算力	功耗	能效比
	TFLOPS	W	TFLOPS/W
Preferred Networks MN-Core[32]	524	500	1.05
本系统	–	–	1.03
云天励飞 DeepEye1000[30,33]	2	2	1.00
地平线 BPU 2 代[34]	2	2	1.00
NVDIA T4[35]	65	70	0.93
华为 Atlas 300T[36]	256	300	0.85
Graphcore C2[39,40]	250	300	0.83
Groq TSP[39]	205	300	0.68
NVIDIA Jetson AGX Xavier[30]	16	30	0.53
谷歌 CloudTPUv3[31]	90	200	0.45
百度昆仑 K200[37,38]	64	150	0.43
NVDIA Tesla V100[42]	125	300	0.42
天数智芯 Iluvatar CoreX I[29]	4.8	13.3	0.36
云燧 T10[28]	80	225	0.36
比特大陆 SOPHON SC5 +[41]	26.4	75	0.35
NVDIA Tesla P100[44]	21.2	300	0.07

通过分析表 2、表 3、图 11 可以得出,基于 ARM + DLP + SRIO 的嵌入式智能计算系统在单纯算力指标上,其表现并不是业界最好的,但是在能效比指标上,处于业界领先地位. 从图 11 来看,能效比最好的是 MN-Core 平台,来自日本 Preferred Networks 研究机构,主要用于模型训练场景,达到 1.05TFLOPS/W. 本系统主要用于模型推理场景,能效比仅次于 MN-Core 平台,达到

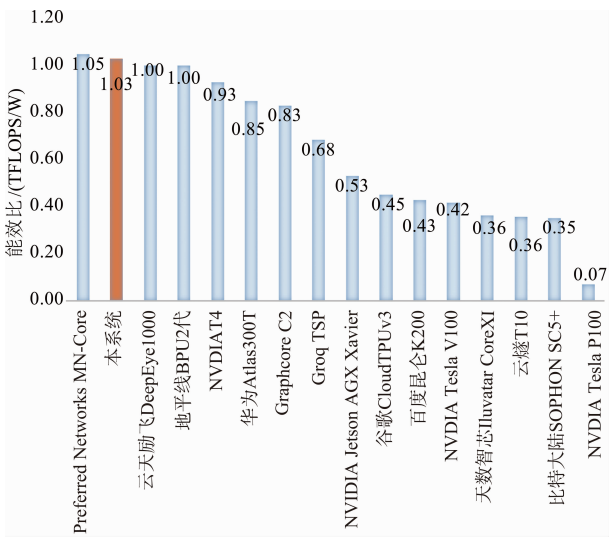


图11 嵌入式智能计算系统与其它AI加速平台的能效比对比

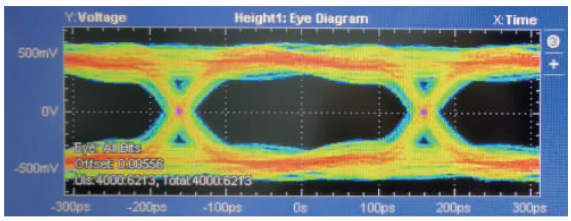
1.03TFLOPS/W,领先于其他 AI 加速平台.

5.4 IO 带宽实验

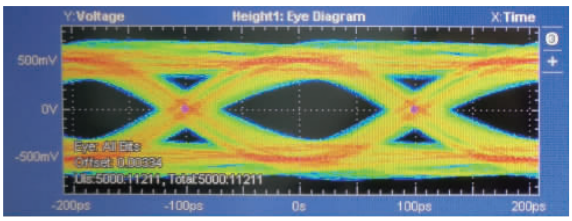
嵌入式智能计算系统对外采用 SRIO 进行高速数据通信. 为了分析本系统的 IO 带宽,本文首先对 SRIO 信号在两种速率模式下的信号完整性进行了测试,捕捉信号眼图,并进行分析;继而采用压力测试的方法对 SRIO 通信带宽进行测试,以获取有效带宽.

图 12(a)、图 12(b)分别是 SRIO 工作在 3.125Gbps、5Gbps 速率档时的信号眼图. 表 4 对比列出了实测眼图参数和 SRIO 标准规范定义的眼图模板参数[43]. 观察眼图测试结果可知,当信号速率为 3.125Gbps 时,眼高(eye height)较高,对噪声的容忍度较好;眼宽(eye width)也较宽,信号总体抖动比较小,信号质量良好. 当信号速率为 5Gbps 时,眼高与眼宽都有一定程度减小,但对比表 4 中的标准眼图模板可知,眼图依然能套住模板. 由于本系统采用的 PCIe-SRIO 桥接芯片最高速率仅支持到 5Gbps,因此对于 SRIO2.0 标准中规定的更高速率(如 6.25Gbps),本系统未进行测试. 对眼图测试结果进

行分析可知,本系统的 SRIO 能够稳定工作在 5Gbps/lane 速率. 由于本系统采用 SRIO 4x 链路,因此本系统的 IO 带宽达到 20Gbps.



(*a*)速率为3.125Gbps



(*b*)速率为5Gbps

图12　SRIO信号眼图

表 4　SRIO 眼图参数实测值与标准规范值对比

参数 / 速率档	眼高		眼幅度		眼宽	
	标准	实测	标准	实测	标准	实测
3. 125Gbps	625mV	628. 98mV	1376mV	1200mV	197. 6ps	244. 52ps
5Gbps	228. 2mV	646. 77mV	1032. 1mV	1500mV	103. 7ps	129. 56ps

6　结论

论文从解决物端及边缘计算的 AI 加速问题出发,分析了嵌入式智能计算系统对 AI 算力、能效比、IO 带宽的特殊需求,提出并研究了基于 ARM + DLP + SRIO 的嵌入式智能计算系统. 该系统综合采用嵌入式 ARM 处理器、DLP 深度学习处理器、SRIO 高速串行总线等技术,设计了适用于物端及边缘 AI 加速的嵌入式异构智能计算架构和高速数据交换通道,从系统层面最大限度保障 AI 算力的发挥. 经测试,本系统 AI 峰值算力达到 114. 9TOPS,能效比达到 1. 03TFLOPS/W,IO 带宽达到 20Gbps. 在智能计算系统领域,其能效比优于国内其它同类板卡或系统,嵌入式环境适应能力优于传统台式机和服务器,可作为物端及边缘环境下智能计算任务的通用硬件加速平台,具有较高的实用价值.

致谢　感谢中国科学院计算技术研究所陈云霁、郭崎、周惠、蔡利军等老师对论文工作的支持和帮助.

参考文献

[1] 彭晓晖,张星洲,王一帆,等. Web 使能的物端计算系统[J]. 计算机研究与发展,2018,55(3):572 – 584.
Peng Xiaohui, Zhang Xingzhou, Wang Yifan, et al. Web enabled things computing system[J]. Journal of Computer Research and Development, 2018, 55(3): 572 – 584. (in Chinese)

[2] 彭晓晖,徐志伟. 基于边缘计算的物端系统挑战与愿景[J]. 中兴通讯技术,2019,25(3):31 – 36,57.
Peng Xiaohui, Xu Zhiwei. Challenges and vision of things system in edge computing[J]. ZTE Technology Journal, 2019,25(3):31 – 36,57. (in Chinese)

[3] 徐志伟,曾琛,朝鲁,彭晓辉. 面向控域的体系结构:一种智能万物互联的体系结构风格[J]. 计算机研究与发展,2019,56(1):90 – 102.
Xu Zhiwei, Zeng Chen, Chao Lu, Peng Xiaohui. Zone-oriented architecture: an architectural style for smart web of everything[J]. Journal of Computer Research and Development, 2019, 56(1): 90 – 102. (in Chinese)

[4] 赵梓铭,刘芳,蔡志平,肖侬. 边缘计算:平台、应用与挑战[J]. 计算机研究与发展,2018,55(2):327 – 337.
Zhao Ziming, Liu Fang, Cai Zhiping, Xiao Nong. Edge computing: platforms, applications and challenges[J]. Journal of Computer Research and Development, 2018, 55(2): 327 – 337. (in Chinese)

[5] Shi W, Cao J, Zhang Q, et al. Edge computing: vision and challenges[J]. Internet of Things Journal, IEEE, 2016, 3(5): 637 – 646.

[6] 施巍松,孙辉,曹杰,张权,刘伟. 边缘计算:万物互联时代新型计算模型[J]. 计算机研究与发展,2017,54(5):907 – 924.
Shi Weisong, Sun Hui, Cao Jie, Zhang Quan, Liu Wei. Edge computing—an emerging computing model for the internet of everything era[J]. Journal of Computer Research and Development, 2017, 54(5): 907 – 924. (in Chinese)

[7] 陈云霁,李玲,李威,郭崎,杜子东. 智能计算系统[M]. 北京:机械工业出版社,2020.

[8] 寒武纪. MLU100 简介[EB/OL]. http://forum.cambricon.com/index.php? m = content&c = index&a = show&catid = 50&id = 127,2019 – 07 – 11.

[9] X Zhang, X Zhou, M Lin, J Sun. ShuffleNet: an extremely efficient convolutional neural network for mobile devices[A]. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition[C]. Piscataway, NJ: IEEE Press,

2018.6848 – 6856.

[10] A G Howard, M Zhu, B Chen, D Kalenichenko, W Wang, T Weyand, M Andreetto, H Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[EB/OL]. https://arxiv.org/abs/1704.04861, 2020-07-27.

[11] 王齐. PCIExpress 体系结构导读[M]. 北京:机械工业出版社,2010.

[12] PCI-SIG. PCI Express Base Specification Revision 3.1a [EB/OL]. https://pcisig.com/specifications/pciexpress/,2015-10-07.

[13] 王勇,林粤伟,吴冰冰. RapidIO 嵌入式系统互连[M]. 北京:电子工业出版社,2006.

[14] RapidIO Trade Association. RapidIO Specification 2.2 [EB/OL]. http://www.rapidio.org/rapidio-specifications/,2011-05.

[15] 刘军伟. 多种高速串行总线的对比研究与分析[J]. 电子测试,2016(3):43 – 45.
Liu Junwei. Comparative study and analysis of multiple high speed serial bus[J]. Electronic Test,2016(3):43 – 45. (in Chinese)

[16] 李超. RapidIO 高速互联技术研究[J]. 现代导航,2017(5):385 – 390.
Li Chao. Research of high-speed RapidIO interconnection technology[J]. Modern Navigation,2017(5):385 – 390. (in Chinese)

[17] 寒武纪. 寒武纪端云一体人工智能开发平台白皮书[EB/OL]. http://forum.cambricon.com/list-79-1.html,2020-07-29.

[18] 寒武纪. 寒武纪软件开发环境[EB/OL]. http://forum.cambricon.com/index.php?m=content&c=index&a=show&catid=50&id=128,2019-07-11.

[19] 刘彦,付彬,李仁发. 高性能嵌入式计算[M]. 北京:机械工业出版社,2016.

[20] 赵二虎,徐勇军,吴济文,安竹林,李超. 用于人工智能处理器的数据互联方法、系统、芯片和装置[P]. 中国专利申请号:201910826850.X,2020-01-17.

[21] 赵二虎,徐勇军,吴济文,李超,安竹林. 面向物端数据处理的嵌入式智能计算机架构[P]. 中国专利申请号:201910605382.3,2020-01-14.

[22] 许家麟,韩思齐,孙宁霄,吴琼之. 硬件加速系统中的 PCIe-SRIO 桥技术[J]. 电子设计工程,2017,25(15):189 – 193.
Xu Jialin, Han Siqi, Sun Ningxiao, Wu Qiongzhi. PCIe-RapidIO bridge for hardware acceleration systems[J]. Electronic Design Engineering,2017,25(15):189 – 193. (in Chinese)

[23] 李红兵. Linux 系统下 PCIE to RapidIO 桥驱动设计与实现[J]. 雷达与对抗,2018,38(2):55 – 58,68.
Li Hongbing. Driver design and implementation of PCIE to RapidIO bridge under Linux operating system[J]. Radar & ECM,2018,38(2):55 – 58,68. (in Chinese)

[24] K He, X Zhang, S Ren, et al. Deep residual learning for image recognition[A]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)[C]. Piscataway, NJ: IEEE Press,2016.770 – 778.

[25] P Molchanov, S Tyree, T Karras, T Aila, J Kautz. Pruning convolutional neural networks for resource efficient inference[A]. Proceedings of the 5th International Conference on Learning Representations [C]. Toulon, France: ICLR,2017.

[26] K Simonyan, A Zisserman. Very deep convolutional networks for large-scale image recognition[A]. Proceedings of the 3rd International Conference on Learning Representations[C]. San Diego, CA, USA: ICLR,2015.

[27] J Redmon, A Farhadi. Yolov3: An Incremental Improvement [EB/OL]. https://arxiv.org/abs/1804.02767, 2018-04-08.

[28] 燧原科技. 云燧 T10 人工智能训练加速卡[EB/OL]. http://site-static.enflame-tech.com/enflame_back/public/uploads/c_video_img/%E4%BA%91%E7%87%A7T10%E5%8D%95%E9%A1%B52020-4-7.pdf, 2020-07-29.

[29] 天数智芯. 边缘计算系统板硬件说明书 v1.0.1[EB/OL]. http://iluvatar.ai/filedownload/34977, 2019-11-19.

[30] 中国人工智能产业发展联盟. AI 芯片技术选型目录[EB/OL]. http://www.199it.com/archives/1097964.html,2020-07.

[31] Paul Teich. Tearing Apart GOOGLE's TPU 3.0 AI Coprocessor [EB/OL]. https://www.nextplatform.com/2018/05/10/tearing-apart-googles-tpu-3-0-ai-coprocessor,2018-05-10.

[32] Preferred Networks. MN-Core Accelerator for Deep Learning[EB/OL]. https://projects.preferred.jp/mn-core/en/#mn-core,2020-05-06.

[33] 云天励飞. 8/16 路视频智能分析边缘模组方案[EB/OL]. https://www.intellif.com/int/product/list18.html,2021-02-24.

[34] 地平线. 自动驾驶边缘 AI 芯片 Journey 征程[EB/OL]. https://www.horizon.ai/product/journey,2020-05-06.

[35] NVIDIA. NVIDIA T4 Tensor Core GPU[EB/OL]. https://www.nvidia.cn/content/dam/en-zz/Solutions/Data-Center/tesla-t4/t4-tensor-core-datasheet.pdf, 2018-10-18.

[36] 华为. Atlas 300T 训练卡规格参数[EB/OL]. https://e.huawei.com/cn/products/cloud-computing-dc/atlas/atlas-300t-training-9000,2021-02-24.

[37] 包永刚. 百度自研 AI 芯片昆仑解读[EB/OL]. http://www. elecfans. com/d/1196758. html,2020-04-03.
[38] 百度智能云. 百度昆仑 AI 加速卡[EB/OL]. https://cloud. baidu. com/product/kunlun. html,2020-05-06.
[39] L Gwennap. Groq Rocks Neural Networks Startup Creates New Architecture:One Core,1,000 TOPS[EB/OL]. https://www. linleygroup. com/mpr/article. php? id = 12245,2020-05-06.
[40] Z Jia,B Tillman,M Maggioni,et al. Dissecting the graphcore IPU architecture via microbenchmarking[EB/OL]. https://arxiv. org/abs/1912. 03413v1,2019-10-07.
[41] 比特大陆. Sophon AI 计算加速卡 SC5 +[EB/OL]. https://sophon. cn/product/introduce/sc5-plus. html,2020-05-06.
[42] NVIDIA. NVIDIA V100 Tensor Core GPU Datasheet [EB/OL]. https://www. nvidia. cn/content/dam/en-zz/zh_cn/Solutions/Data-Center/tesla-v100/volta-v100s-datasheet-a4-nvidia-1209899-r1-web-zhcn. pdf,2020-01.
[43] 刘升财. ATCA 平台上的 RapidIO 链路设计与分析[D]. 成都:电子科技大学,2014.
[44] NVIDIA. NVIDIA Tesla P100 GPU Accelerator Datasheet [EB/OL]. https://www. nvidia. cn/content/dam/en-zz/Solutions/Data-Center/tesla-p100/pdf/nvidia-tesla-p100-datasheet. pdf,2016-10-16.

作者简介

赵二虎　男,1985 年生于河北邢台. 现为中国科学院计算技术研究所高级工程师,专项技术研究中心智算平台研究组组长,在读博士研究生,主要研究方向为嵌入式智能计算系统.
E-mail:zhaoerhu@ ict. ac. cn

吴济文　男,1987 年生于江西上饶. 现为中国科学院计算技术研究所工程师,主要研究方向为嵌入式智能计算系统的软硬件协同优化.
E-mail:wujiwen@ ict. ac. cn

查晶晶　女,1994 年生于河南周口. 现为中国科学院计算技术研究所工程师,主要研究方向为边缘异构智能计算系统中的算法优化与加速.
E-mail:zhajingjing@ ict. ac. cn

郭　振　男,1992 年生于陕西商洛. 现为中国科学院计算技术研究所工程师,主要研究方向为嵌入式智能计算系统的算法移植与优化.
E-mail:guozhen@ ict. ac. cn

徐勇军　男,1979 年生于安徽安庆. 现为中国科学院计算技术研究所正研级高级工程师、博士生导师、专项技术研究中心主任,主要研究方向为数据智能.
E-mail:xyj@ ict. ac. cn