计算机研究与发展
Journal of Computer Research and Development
https://crad.ict.ac.cn
63(1): 1−14, 2026

本期推荐

面向分布式计算的类脑智能处理器指令集架构设计

冯 烁 路冬冬 尹 飞 杨剑新 班冬松 何 军 颜世云 李 媛 睢浩宇
(上海高性能集成电路设计中心 上海 201204)
(fengshuo_1010@163.com)

Distributed Computing-Oriented ISA Design for Brain-Inspired Intelligence CPU

Feng Shuo, Lu Dongdong, Yin Fei, Yang Jianxin, Ban Dongsong, He Jun, Yan Shiyun, Li Yuan, and Ju Haoyu
(*Shanghai High Performance Integrated Circuit Design Center*, *Shanghai* 201204)

Abstract As one of the most distinctive representations for distributed computing, an end-edge-cloud collaborative computing system is capable of effectively bringing such applications as Internet of things, large-scale language models, and digital twins into real life. By mimicking the neural activities that take place in the human brain, brain-inspired computing has various advantages, including energy efficiency, high speed, high error-tolerance, and desirable scalability. Through leveraging the event-driven mechanism and the sparsity in spike generation of spiking neural networks, the real-time processing capability and energy efficiency of an end-edge-cloud collaborative computing system can be significantly improved. In this paper, a distributed computing-oriented ISA (instruction set architecture) design for brain-inspired intelligence CPU is studied. Bearing in mind the requirements of delay-sensitive, low-power, and high diversity for end devices, we focus on the software-hardware interface, i.e., ISA, and propose a hardware design that is rooted in the current systems, easy to implement and upgrade, safety-aware and self-controllable, and compatible with heterogeneous architectures. Along with a corresponding CPU micro-architecture design, a dozen of instructions specifically conceived for brain-inspired computing are proposed based on a well established ISA, which sets the stage for empowering distributed computing systems with brain-inspired computing. We firmly believe that this paper has discovered a fruitful research field, and sincerely hope that more interests can be stimulated in ISA designs for brain-inspired intelligence CPU in both academic and industrial communities.

Key words distributed computing; brain-inspired intelligence; spiking neural network; instruction set architecture (ISA); CPU micro-architecture; neuromorphic chips

摘 要 作为分布式计算的典型体现之一,端边云协同计算系统能够有效推动物联网、大模型、数字孪生等人工智能技术的垂直落地应用。类脑计算是一种受大脑工作方式启发而提出的智能计算技术,具有能效高、速度快、容错度高、可扩展性强等优点。通过利用脉冲神经网络的事件驱动机制和脉冲稀疏发放等特性,类脑计算能够极大地提升分布式端边云系统的实时处理能力和能量效率。针对分布式终端设备的高实时、低功耗、强异构等特点,聚焦于指令集架构这一软硬件的交互界面,给出了一种立足现有系统、易于部署升级、安全自主可控、异构融合兼容的硬件设计方案,一共提出了 12 条类脑计算指令,完成了基于某国产指令系统的类脑指令集和对应微结构的定制化设计,为类脑计算赋能分布式计算系统奠定了技术基础。

关键词 分布式计算;类脑智能;脉冲神经网络;指令集架构;处理器微结构;神经拟态芯片

中图法分类号 TP338.8

收稿日期:2024-08-13;**修回日期**:2025-04-22

通信作者:冯烁(fengs13@mcmaster.ca)

DOI: 10.7544/issn1000-1239.202440675 CSTR: 32373.14.issn1000-1239.202440675

1 研究背景与设计意义

1.1 端边云协同计算系统

分布式计算是一种将大型计算任务拆解到离散的小型计算资源上进行并行处理的计算范式,可以在一定范围内的计算节点上实现负载平衡。通过资源的分时共享,能够显著提高计算效率和资源利用率,同时增强系统级的可拓展性和容错能力。随着物联网、大模型、数字孪生等应用的快速发展,传统的超级计算机、计算网络和数据中心之间的界限越来越模糊,这一趋势为异构算力网、超算互联网等新型算力基础设施的构建提出了巨大的需求,也推动着分布式计算不断发挥出更大的潜能[1]。

作为分布式计算的典型体现之一,端边云协同计算系统是一种集成了云计算平台、边缘计算网络和终端设备的融合计算架构,旨在通过三者的密切协作和相互配合,实现泛在数据的优化处理和新型应用的效能提升[2]。

在端边云系统中,云端作为数据处理中心,拥有丰富的计算资源和存储资源,负责接收来自边缘服务器的海量数据,能够提供连续、稳定和按需访问的服务,但实时性和隐私性相对较差;边缘网络或设备将计算服务拓展到了网络边缘,负责收集和转发终端数据,能够在压缩数据量的同时实现信息提取,避免了超长网络传输延迟和带宽过度占用,但由于边缘网络的资源受限,难以对大量不同类型的终端用户提供差异化和个性化的接入服务;终端设备作为末梢节点,负责原始数据的采集、处理、消费和上传,能够提供无处不在和高移动性的服务覆盖,且具有较高的实时性,但由于终端设备的功能薄弱、功耗敏感和硬件异构等特性,边缘网络对其提供的服务质量、传输可靠性和用户服务体验(quality of service, QoS)通常难以保证。

由以上分析可知,为了真正实现端边云协同计算系统的融合优势,不仅需要提升全网算力调度和垂直管控能力,也需要提升三者之间的耦合度和适配度;不仅需要系统级和网络级的顶层设计优化,也需要节点级和硬件级的底层设计创新。

1.2 类脑计算技术

类脑计算是一种受大脑工作方式启发而提出的智能计算技术,以脑科学、认知科学、计算神经学等领域的研究成果为基础,通过学习和借鉴生物神经系统的信息处理机制、生理结构和主要功能,追求在人工计算系统中模拟和实现生物大脑的处理能力和处理效率,具有能效高、速度快、容错度高、可扩展性强等优点[3-4]。

在类脑计算领域,脉冲神经网络(spiking neural network, SNN)作为能够高效表征生物神经系统的信息处理机制、结构和功能的一个典型代表,逐步成为了研究焦点,是当前类脑计算研究中最为常用的人工神经网络模型之一。SNN 同时具备较高的生物可靠性和计算有效性,通过模仿生物神经元的脉冲释放行为进行信息传递,并利用脉冲发放的时间和频率来编码信息。由于其具有受稀疏事件驱动的特性,SNN 网络的能量效率很高,并且具有强实时性,非常适合于边缘计算和终端设备等对功耗和延时敏感的应用场景[5]。因此,类脑计算技术能够极大地提升端边云协同计算系统的整体效能[6]。

具体而言,在分布式计算系统中运用类脑计算技术的收益主要包括:1)类脑计算的事件驱动机制,能够对终端设备发起的本地事件进行快速响应,从而大幅提升分布式端边云系统的实时处理能力;2)类脑计算的数据处理对象是易于编码的脉冲序列,能够减少需要传输的数据量,从而提升网络带宽的利用效率,缓解终端设备和边缘网络的带宽压力;3)类脑计算的神经元脉冲发放具有稀疏性,由此带来的低功耗优势,能够显著提升协同计算系统中的各类计算节点,尤其是边缘计算和终端设备的能量效率。

1.3 论文工作和主要贡献

本文从分布式计算和端边云协同计算系统的整体应用效能出发,聚焦于终端设备的高实时、低功耗、强异构等特点,提炼其硬件层面的设计优化需求,提出利用类脑计算技术进行底层创新的方法,完成面向分布式计算的类脑指令集和对应微结构设计,给出了一种立足现有系统、易于部署升级、安全自主可控、异构融合兼容的硬件设计方案。

由于现有的分布式计算系统大多对深度神经网络(deep neural network, DNN)等常见的人工神经网络提供了软硬件层面的支持,而 SNN 可以视为包含了神经动力学特性的 DNN 网络,因此,从硬件角度来看,只需要进行增量式的设计支持,基于现有系统的硬件架构和接口设计,增加类脑专用指令和类脑处理单元等功能部件,即可实现 2 类计算模式的功能

兼容[7]。

指令集架构(ISA)作为软件和硬件之间的接口，定义了一套供程序员使用的指令集及其执行方式，以及处理器支持的操作类型、寄存器的数量和功能、内存访问方式等。指令集架构的权衡取舍和设计优劣，将直接影响计算系统的整体处理效率。端边云协同计算系统中的指令集设计，需要根据分布式计算的特定应用需求，重点关注计算并行性、功能完备性、实现复杂度等因素。

因此，如果分布式系统中的终端硬件设备能够支持类型丰富的指令集，那么边缘网络和云侧数据中心就可以采用更为通用的软件工具和模型算法，进而在通用软件支撑环境和异构硬件平台之间搭建起一座桥梁，从而降低分布式计算系统的部署和升级难度，高效推进端边云系统的协同演进[8]。

本文的主要贡献包括2个方面：

1)针对端边云协同计算系统和类脑计算技术的特点，完成了基于某国产指令系统的定制化设计，一共提出了12条类脑计算专用指令，丰富和增强了国产自主指令系统，为类脑计算赋能分布式计算系统奠定了技术基础；

2)针对当前主流类脑处理器设计中存在的安全可控风险和信息交互代价较大等不足，提出了一种基于某国产处理器核心的细粒度融合兼容的类脑微结构设计方案，既能够实现完全的自主可控，又避免了频繁的信息交互，有助于提升分布式计算系统的资源利用率和安全可靠性。

2 类脑智能处理器指令集设计现状

本节对具有代表性的HiPU系列芯片和达尔文系列芯片所设计研发的类脑指令集，分别进行简要的分析和介绍。

2.1 HiPU系列芯片指令集介绍

HiPU系列芯片是由西安交通大学研发的神经网络加速器，分别面向FPGA与ASIC的2种形态实现。其中，面向FPGA实现的有HiPU100, HiPU300, HiPU151等设计，面向ASIC实现的有HiPU200和HiPU400等设计。此外，另有多款新型设计目前处于研发阶段，HiPU系列芯片正在不断迭代发展[9]。

以HiPU200芯片为例，该芯片的指令集基于标准RISC-V指令集设计，其特点是在标量指令集的基础上，扩展了面向神经网络运算的高效向量指令和矩阵指令。例如，向量聚合指令(vector fused instruction, VFI)的目的是完成神经网络的特征图在每个卷积层结束时所需实现的偏置加和以及量化，向量重排指令(vector align instruction, VAI)可以实现向量寄存器中各个元素的位置交换，而向量查表指令(vector look-up instruction, VLI)旨在实现神经网络的激活函数所需的多种映射方式等。

2023年，在支持DNN的HiPU200指令集基础上，针对DNN和SNN二者的共性特点和个性差异，HiPU研发团队进一步提出了支持SNN的类脑指令集。具体而言，提出了适合SNN计算模型的脉冲序列存取指令(LDS和SDS指令)，实现了脉冲数据在片外存储器和片上存储器之间的高效传输；提出了脉冲序列压解指令(PACK和UNPACK指令)，实现了脉冲数据的解压运算、压缩传输和存储，提高了带宽和存储空间的利用率；提出了脉冲序列运算指令(GP和CLS指令)，实现了膜电位累积、脉冲信号生成等SNN专有计算需求的特殊设计。

该类脑指令集的设计提出，为类脑智能芯片提供了一套简洁高效的指令集，能够模拟生物神经网络的工作机理和动力学行为，支持常见的类脑算法，并在硬件级别上优化计算和存储等操作，同时满足高度生物相似性的功能要求。

2.2 达尔文系列芯片指令集介绍

2015年，浙江大学发布了第1代达尔文类脑芯片，最多可以支持2 048个神经元，包含超过400万个神经突触，是国内首款能够支持SNN计算的神经拟态芯片[10]。

2019年发布的第2代达尔文芯片，采用了优化的LIF神经元模型和不同精度的神经突触，每个芯片包含约15万个神经元、1 000万个神经突触，支持芯片级扩展，并采用了全局异步、局部同步的芯片级设计架构。

2022年发布的第3代达尔文芯片，具备在线学习功能，包含235万个神经元、超过1亿个神经突触，并且采用了RISC-V异构融合的设计，在神经元规模、突触连接密度、片上学习能力、峰值计算能力和平均能耗等方面，都取得了较大提升[11]。尤其值得注意的是，第3代达尔文芯片设计实现了面向脉冲神经元计算特点的领域专用指令集架构，能够支持灵活可配置的神经元编程模型和片上学习规则设计，并且能够实现参数读取和状态更新等操作的高度并行。该指令集由10条基本指令及若干扩展指令构成。其基本指令可被分为3组：第1组侧重于装入和存储操作，包括用于读写状态变量的LSIS和LSLS指令，

以及用于读取参数的 LDIP 和 LDLP 指令; 第 2 组侧重于状态变量的更新, 包括 UPTIS, UPTVM, UPTLS, UPTWT, UPTTS 等指令, 分别用于更新和调整状态变量、膜电位值、学习变量、突触权重、临时变量等; 第 3 组包含 1 条 GSPRS 指令, 专门用于完成脉冲生成的操作。此外, 该指令集的扩展指令还包含少量的 RISC 架构常见指令, 通过使用乘法器、移位器和 LUT 等基本计算单元, 能够实现指数运算和除法等操作。该指令集的核心原则是基于 SNN 网络的计算特点, 将常用操作合并到单个指令中。通过这种方式, 不仅减少了指令所需的内存使用, 还将计算过程中指令解码所需的时间降至最低。同时, 该指令集能够构建和支持多种形式的 SNN 计算模型, 从而实现了指令集设计的高度灵活性和有效性。

2.3 不同架构类型的分析对比

除了 HiPU 系列芯片和达尔文系列芯片之外, 还有一些具有代表性的类脑智能芯片也支持了类脑计算指令的定制化设计。例如, 曼彻斯特大学研发的 SpiNNaker 多核处理器芯片, 通过基于 ARM 核心的指令设计, 实现了灵活可配置的神经元模型[12-13]; 英特尔公司研发的 Loihi2 类脑芯片, 采用一系列微码指令对单个可编程神经元的行为进行描述, 并提出了包括脉冲生成指令 SPIKE 在内的多条类脑专用指令[14]。

尽管如此, 当前面向 SNN 网络的专用类脑指令的研究成果仍然十分有限, 多数类脑指令集的设计均处于初期阶段, 尚未形成规范成熟的设计标准, 仍然需要不断深入的探索研究。

从架构角度分析, HiPU 系列芯片和达尔文系列芯片等, 均采用了主从异构的架构类型: 类脑芯片作为加速器部分, 负责特定的类脑计算任务; 通用处理器作为主控部分, 负责一般性的控制和管理任务。对于此类基于主从架构的设计, 其主控部分一般采用成熟的通用指令集, 例如 ARM 指令集或 x86 指令集等; 其加速器部分一般采用开源、灵活、易于迭代开发的指令集, 例如 RISC-V 指令集等。

与主从异构形式的类脑计算指令集不同, 本文提出一种基于某国产指令系统的类脑指令集设计方案。该方案采用融合兼容的架构类型, 即主控部分和类脑加速部分均采用国产自主指令集进行设计。相关类脑计算芯片的指令系统对比如表 1 所示。

在本文所提方案中, 主控部分和类脑加速部分的通用功能, 均可由当前的国产指令系统进行支持; 而类脑加速部分的专用功能, 则需要在该指令系统的现有基础上, 新增特定功能的新指令进行支持, 并在扩展指令系统中加以实现。具体如下所述。

Table 1 Comparison of ISA Between Different Brain-Inspired Computing Chips

表 1 不同类脑计算芯片的指令集架构对比

对比项	HiPU 系列芯片	达尔文系列芯片	所提类脑芯片设计
架构类型	主从异构	主从异构	融合兼容
主控部分	ARM 指令集	ARM/x86 指令集	某国产自主指令集
类脑加速部分的通用功能	RISC-V 标准指令集	RISC-V 标准指令集	某国产自主指令集
类脑加速部分的专用功能	RISC-V 扩展指令集	RISC-V 扩展指令集	某国产自主指令集拓展

3 基于国产自主指令系统的类脑计算指令集设计

3.1 某国产自主指令系统概述

本文所采用的某国产自主指令系统为 64 b 字长的 Load/Store 型 RISC 架构, 所有指令均采用定长的 32 b 格式, 支持 8 b, 16 b, 32 b, 64 b 整数运算, 也支持部分 256 b 整数运算。支持 IEEE-754 32 b 单精度浮点(FP32, 称为 S-浮点)和 64 b 双精度浮点(FP64, 称为 D-浮点)运算, 支持 2 种 16 b 半精度浮点(BF16 和 FP16, 统称为 H-浮点), 仅支持半精度浮点与单精度浮点之间的转换运算。支持 256 b 单指令流多数据流(SIMD)的短向量整数和浮点运算。

该国产自主指令系统包含基本指令系统和扩展指令系统, 支持具有短向量特征的 SIMD 数据处理。运算指令包括整数运算指令和浮点运算指令, 运算一般均在寄存器之间进行, 结果仍然写回寄存器; 存储器访问指令包括存储器装入和存储以及同步互斥和锁指令, 实现寄存器和存储器之间的数据传送; 指令系统还包括转移指令在内的控制指令、特权指令和实现整数寄存器与浮点寄存器之间数据传递指令等。

该指令系统具有 5 种基本指令格式, 分别为系统调用指令格式、转移指令格式、存储器指令格式、简单运算指令格式和复合运算指令格式, 分别使用 0, 1, 2, 3, 4 个寄存器。所有格式的指令长度均为 32 b, 其中操作码长度为 6 b, 每个寄存器域占用 5 b(对应 32 个整数、浮点或向量寄存器)。其基本指令格式, 如表 2 所示。

3.2 支持 SNN 的类脑指令设计

在该国产自主指令系统的现有基础上, 针对 SNN 网络在膜电位累积(时域积分)、脉冲生成(时域发放), 以及数据存取(空域传送)等方面的计算特点,

Table 2 Basic ISA Format for A Domestic Processor (Neuromorphic Extension Included)

表 2 某国产处理器的基本 ISA 格式（含类脑扩展）

指令格式的操作码与功能码						指令格式
[31:26]	[25:21]	[20:16]	[15:10]	[9:5]	[4:0]	
操作码	功能码	功能码	功能码	功能码	功能码	系统调用指令格式
操作码	*Ra*	*disp*	*disp*	*disp*	*disp*	转移指令格式
操作码	*Ra*	*disp*	*disp*	*disp*	*disp*	存储器指令格式
操作码	*Ra*	*Rb*	功能码	功能码	*Rc*	简单运算指令格式
操作码	*Ra*	*Rb*	功能码	*Rc*	*Rd*	复合运算指令格式
操作码	*Ra*	*Rb*	功能码，*Rc*	*Rd*	*Re*	类脑运算指令格式（新增）

本文完成了面向 SNN 网络的定制化设计，提出了 10 条支持 SNN 的类脑计算专用指令，具体如下。

3.2.1 膜电位累积（时域积分）指令

1）膜电位单输入累积基本指令（SELEQ_S）

常见 DNN 神经元模型的膜电位更新过程，可以看作是“乘累加”计算；与之不同，SNN 神经元模型的膜电位更新机制，可以看作“有条件的累加”计算。因此，针对 SNN 神经元的膜电位累积特点，可以新增一条类别为整数条件选择的指令，作为专用的膜电位单输入累积基本指令，如表 3 所示。

Table 3 Features of SELEQ_S

表 3 SELEQ_S 的特征

寄存器格式操作码与功能码		描述	指令格式
[12:10]	[15:13]		
0x11.0	0x1	整数等于“0”时选择累积	整数复合运算指令格式

SELEQ_S 的相关定义如下。

①助记符：SELEQ_S。

/*整数等于“0”时选择累积*/

②格式：SELEQ_S *Ra.rl*, *Rb.rl*, *Rc.rl*, *Rd.wl*。

③操作：IF TEST(*Rav*, EQ) THEN

Rd ← *Rbv*;

ELSE *Rd* ← *Rbv*+*Rcv*;

ENDIF

④指令描述：对 *Ra* 进行条件测试，若 *Ra*＝0，则将 *Rb* 的值写入 *Rd*，否则将 *Rb* 的值与 *Rc* 的值之和写入 *Rd*。

⑤含义说明：*Ra* 中存放输入脉冲值，若 *Ra*＝0，无需累加膜电位，将此前的膜电位累积值（*Rb* 的值）写入 *Rd*；否则，需要累加膜电位，将此前的膜电位累积值（*Rb* 的值）与当前的突触权重值（*Rc* 的值）之和写入 *Rd*。指令执行完毕后，*Rd* 的值表示当前的膜电位累积值。

此外，值得注意的是，该指令的功能，仅使用 3 个寄存器也可以实现：根据 *Ra* 的值，选择保持 *Rb* 的值，或者选择将 *Rb* 的值与 *Rc* 的值之和写入 *Rb*。通过这种方式，可以节省 1 个物理寄存器，节约硬件资源。从另一个角度来说，如果仍然使用 4 个寄存器，则可以实现同时对 2 个神经元输入进行膜电位累积，即为膜电位双输入累积基本指令。

2）膜电位双输入累积基本指令（SELEQ_D）

对于神经元连接度较高的 SNN 网络，前向神经元的个数一般较多，为了提高神经元膜电位的累积效率，可以进一步新增 1 条整数条件选择指令作为专用的膜电位双输入累积基本指令。

SELEQ_D 的相关定义如下。

①助记符：SELEQ_D。

/*2 个整数等于“0”时分别选择累积*/

②格式：SELEQ_D *Ra.rl*, *Rb.rl*, *Rc.rl*, *Rd.wl*。

③操作：*temp* ← 0;

IF TEST(*Rav*[0], EQ) THEN

temp ← *temp*;

ELSE *temp* ← *temp*+*Rbv*;

ENDIF

IF TEST(*Rav*[1], EQ) THEN

temp ← *temp*;

ELSE *temp* ← *temp*+*Rcv*;

ENDIF

Rd ← *temp*+*Rdv*。

④指令描述：对 *Ra*[0] 进行条件测试，若 *Ra*[0] 不为 0，则将 *Rb* 的值写入 *temp*；对 *Ra*[1] 进行条件测试，若 *Ra*[1] 不为 0，则将 *temp* 与 *Rb* 的值之和写入 *temp*；将 *temp* 与 *Rd* 的值之和写入 *Rd*。即，对 *Ra*[0] 和 *Ra*[1] 分别进行条件测试，分别选择是否将 *Rb* 的值与 *Rc* 的值累加写入 *Rd*。

⑤含义说明：*Ra*[0] 中存放第 0 个输入脉冲值，若 *Ra*[0]＝0，无需累加膜电位，否则需要累加；*Ra*[1] 中存放第 1 个输入脉冲值，若 *Ra*[1]＝0，无需累加膜电位，否则需要累加。将当前需要累加的突触权重值（共有 0，1，2 个共 3 类情况），与此前的膜电位累积值（*Rd* 的值）之和写入 *Rd*。指令执行完毕后，*Rd* 的值表示当前的膜电位累积值。

3）膜电位单组输入累积 SIMD 指令（VSELLBCW_S）

为了提升类脑指令密度和提高 SNN 运算效率，可以通过新增 SIMD 拓展指令实现 SNN 神经元模型

的膜电位更新的批量处理,完成“有条件的批量累加”计算。

其指令格式,可采用浮点复合运算指令格式的寄存器格式,由 6 b 功能码确定具体指令功能,如表 4 所示。

Table 4 Register Format of Instruction Format for FP Complex Computation (Neuromorphic SIMD Extension Included)

表 4 浮点复合运算指令格式的寄存器格式(含类脑 SIMD 扩展)

指令格式的操作码与功能码						指令格式
[31:26]	[25:21]	[20:16]	[15:10]	[9:5]	[4:0]	
操作码	***Fa*/*Va***	***Fb*/*Vb***	功能码	***Fc*/*Vc***	***Fd*/*Vd***	浮点复合运算指令格式的寄存器格式
操作码	***Fa*/*Va***	***Fb*/*Vb***	功能码, ***Fc*/*Vc***	***Fd*/*Vd***	***Fe*/*Ve***	类脑 SIMD 运算指令格式(新增)

由此,新增 1 条类别为整数向量条件选择的指令作为专用的膜电位单组输入累积 SIMD 指令。该指令的相关定义如下。

①助记符: VSELLBCW_S。

/*字向量低位为“0”时选择累积*/

②格式: VSELLBCW_S ***Va*.*rw***, ***Vb*.*rw***, ***Vc*.*rw***, ***Vd*.*ww***。

③指令描述: 字向量低位为“0”时选择累积。指令的具体含义为, ***Va***, ***Vb***, ***Vc***, ***Vd*** 均为存放 8 个字整数的向量寄存器, *Va*[32×*i*](0≤*i*≤7)中存放 8 个输入脉冲值,若 *Va*[32×*i*]==0(0≤*i*≤7),无需累加膜电位,将此前的膜电位累积值(***Vb*** 的值)写入 ***Vd*** 的对应位置;否则,需要累加膜电位,将此前的膜电位累积值(***Vb*** 的值)与当前的突触权重值(***Vc*** 的值)之和写入 ***Vd*** 的对应位置。指令执行完毕后, ***Vd*** 的值表示当前的 8 个膜电位累积值。

与 SELEQ_S 指令类似, VSELLBCW_S 指令的功能仅使用 3 个向量寄存器也可以实现: 根据 ***Va*** 的值,选择保持 ***Vb*** 的值,或者选择将 ***Vb*** 的值与 ***Vc*** 的值之和写入 ***Vb***。通过这种方式,可以节省 1 个物理向量寄存器,节约硬件资源。从另一个角度来说,如果仍然使用 4 个向量寄存器,则可以实现同时对 2 组神经元输入进行膜电位累积,即为膜电位双组输入累积 SIMD 指令。

4)膜电位双组输入累积 SIMD 指令(VSELLBCW_D)

为了提高神经元膜电位的累积效率,可以进一步新增一条整数向量条件选择指令,作为专用的膜电位双组输入累积 SIMD 指令。

该指令的相关定义如下。

① 助记符: VSELLBCW_D。

/*字向量低 2 位为“0”时分别选择累积*/

② 格式: VSELLBCW_D ***Va*.*rw***, ***Vb*.*rw***, ***Vc*.*rw***, ***Vd*.*ww***。

③ 操作:

```
FOR (i=0, i<8, i++)
    temp ← 0;
    IF TEST (Va[32×i], EQ) THEN
        temp ← temp;
    ELSE temp ← temp + Vbv[32×i+31: 32×i];
    ENDIF
    IF TEST (Va[32×i+1], EQ) THEN
        temp ← temp;
    ELSE temp ← temp + Vcv[32×i+31: 32×i];
    ENDIF
    Rd ← temp + Rdv[32×i+31: 32×i];
ENDFOR
```

与 VSELLBCW_S 指令不同, VSELLBCW_D 指令的操作可描述为: 字向量低 2 位为“0”时分别选择累积。指令的具体含义为: *Va*[32×*i*](0≤*i*≤7)中存放 8 个输入脉冲值,若 *Va*[32×*i*]==0(0≤*i*≤7),无需累加膜电位,将此前的膜电位累积值(***Vb*** 的值)写入 ***Vd*** 的对应位置;否则,需要累加膜电位,将此前的膜电位累积值(***Vb*** 的值)与当前的突触权重值(***Vc*** 的值)之和写入 ***Vd*** 的对应位置。*Va*[32×*i*+1](0≤*i*≤7)的含义,与之类似。

关于按组累积,也可作如下理解: *Va*[0] 中存放第 0 组第 0 个输入脉冲值,若 *Va*[0]=0,无需累加膜电位,否则需要累加; *Va*[1] 中存放第 0 组第 1 个输入脉冲值,若 *Va*[1]=0,无需累加膜电位,否则需要累加。从 *Va*[32] 和 *Va*[33](第 1 组)直至 *Va*[224] 和 *Va*[225](第 7 组),以此类推。将每组当前需要累加的突触权重值与此前的膜电位累积值(***Vd*** 的值)之和写入 ***Vd***。指令执行完毕后,完成双组输入的累积, ***Vd*** 的值表示当前的 8 个膜电位累积值。

3.2.2 脉冲生成(时域发放)指令

1)脉冲生成基本指令(SELGT_S5R)

SNN 网络采用脉冲进行信息编码,脉冲信号是一种二值信号,信号为“1”代表神经元被激活(产生 1 个事件)并传递至其后端连接的神经元,信号为“0”代表神经元未被激活且未产生脉冲事件。SNN 神经元模型的脉冲生成机制体现于神经元的胞体动力学:当神经元的膜电位累积值超过某个阈值时,胞体就

会发放 1 个脉冲；否则，不进行脉冲发放。因此，针对 SNN 神经元的脉冲生成特点，结合某国产指令系统的操作码使用情况，本文给出 2 种脉冲生成基本指令的设计方法，并分别进行介绍。

该国产指令系统共有 64 个操作码，如果将其中一个操作码“0x1F”专门用于类脑计算，则可以实现对脉冲生成运算的高效支持。

如表 2 所示，当前该国产处理器有 5 种基本指令格式，分别使用 0, 1, 2, 3, 4 个寄存器。为了实现对类脑脉冲生成运算的专项支持，可以定义一种新的基本指令格式并使用 5 个寄存器，称为“类脑运算指令格式”。

类脑运算指令格式的指令长度仍然保持为 32 b，其中操作码的长度为 6 b，5 个寄存器域各占用 5 b，上述已占用位数共计 31 b。因此，仅剩余 1 b 可用于功能码。在本文中，将第 [15] 位定义为功能码，并用于区分脉冲生成基本指令（SELGT_S5R）（第 [15] 位为 0x0）和脉冲生成 SIMD 指令（第 [15] 位为 0x1），如表 5 所示。

Table 5　Features of SELGT_S5R

表 5　SELGT_S5R 特征

操作码	功能码 [15]	描述	指令格式
0x1F	0x0	整数大于“0”时选择生成脉冲	类脑运算指令格式

由此，占用该国产指令系统“0x1F”操作码的 SELGT_S5R 的相关定义如下。

①助记符：SELGT_S5R。

/*整数大于“0”时选择生成脉冲（包含 5 个操作数）*/

②格式：SELGT_S5R *Ra.rl*, *Rb.rl*, *Rc.rl*, *Rd.wl*, *Re.wl*。

③操作：IF TEST(*Rav*−*Rbv*, GT) THEN

Rd ← *Rcv*, *Re* ← 1;

ELSE *Rd* ← *Rav*, *Re* ← 0;

ENDIF

④指令描述：对 *Ra*−*Rb* 进行条件测试，若 *Ra*−*Rb*>0，则将 *Rc* 的值写入 *Rd* 并将 *Re* 的值置 1；否则将 *Ra* 的值写入 *Rd* 并将 *Re* 的值置 0。

⑤含义说明：*Ra* 中存放膜电位累积值，*Rb* 中存放膜电位阈值，*Rc* 中存放膜电位复位值。若膜电位累积值超过阈值，则将膜电位复位，即将膜电位复位值（*Rc* 的值）写入 *Rd*；并且发放脉冲，即将脉冲值（*Re* 的值）置 1。否则，膜电位无需复位，即将膜电位累积值（*Ra* 的值）写入 *Rd*；且不发放脉冲，即将脉冲值（*Re* 的值）置 0。指令执行完毕后，*Rd* 的值表示当前的膜电位累积值，*Re* 的值表示当前的脉冲生成情况。

2）脉冲生成基本指令（SELGT_S4R）

如果采用不占用该国产指令系统“0x1F”操作码的设计方案，虽然无法对脉冲生成运算进行专项支持，但可以继续保持指令系统的开放性和灵活性。

由于该国产处理器的基本指令格式无法使用 5 个寄存器，因此，复用整数复合运算指令格式的寄存器格式（类脑格式），新增一条操作码与功能码 [15:13] 为 0x3 的整数条件选择指令，作为专用的脉冲生成基本指令。

SELGT_S4R 的相关定义如下。

①助记符：SELGT_S4R。

/*整数大于“0”时选择生成脉冲（包含 4 个操作数）*/

②格式：SELGT_S4R *Ra.ml*, *Rb.rl*, *Rc.rl*, *Rd.wl*。

③操作：IF TEST(*Rav*−*Rbv*, GT) THEN

Ra ← *Rcv*, *Rd* ← 1;

ELSE *Rd* ← 0;

ENDIF

指令描述和含义说明类似可得，故略。指令执行完毕后，*Ra*（既是源操作寄存器，也是目标操作寄存器）的值表示当前的膜电位累积值，*Rd* 的值表示当前的脉冲生成情况。

值得注意的是，由于 SNN 脉冲的稀疏性，绝大部分时间下神经元都不激发，因此，膜电位的复位（即 *Ra* 的复位）是小概率事件。

3）脉冲批量生成 SIMD 指令（VSELLTW_S5R）

与膜电位更新的批量处理相类似，为了提升类脑指令密度和运算效率，可以通过新增 SIMD 拓展指令实现 SNN 神经元脉冲生成的批量处理。

对于 SELGT_S5R，采用的是该国产指令系统基本指令格式中新定义的“类脑运算指令格式”（见表 2）。对于 VSELLTW_S5R，在本文中定义一种新的浮点复合运算指令格式，称为“类脑 SIMD 运算指令格式”，使用 5 个寄存器（见表 4）。

采用占用该国产指令系统“0x1F”操作码的设计方案，新增 1 条整数向量条件选择指令，作为专用的脉冲批量生成 SIMD 指令。

VSELLTW_S5R 的相关定义如下。

①助记符：VSELLTW_S5R。

/*字整数向量小于 0 时选择生成脉冲（包含 5 个操作数）*/

②格式：VSELLTW_S5R ***Va.rw***, ***Vb.rw***, ***Vc.rw***, ***Vd.ww***, ***Ve.ww***。

③操作：

```
FOR (i=0; i<8; i++)
  IF TEST(Vb[32×i+31:32×i]−Va[32×i+31:32×i],
    VSELLTW_S5R) THEN
      Vd[32×i+31: 32×i]←Vcv[32×i+31: 32×i],
        Ve[32×i]←1;
    ELSE Vd[32×i+31: 32×i]←Vav[32×i+31: 32×i],
      Ve[32×i]←0;
    ENDIF
  ENDFOR
```

④指令描述：***Va***, ***Vb***, ***Vc***, ***Vd*** 均为存放 8 个字整数的向量寄存器，测试 ***Vb***−***Va*** 中每个字元素，如果 ***Vb***−***Va*** 中每个字元素小于 0，则将向量 ***Vb*** 对应的字元素写入 ***Vd*** 的对应位置，并将 ***Ve*** 对应的字元素最低位置 1；否则将向量 ***Vc*** 对应的字元素写入 ***Vd*** 的对应位置，并将 ***Ve*** 对应的字元素最低位置 0。

⑤含义说明：***Va*** 中存放 8 个膜电位累积值，***Vb*** 中存放 8 个膜电位阈值，***Vc*** 中存放 8 个膜电位复位值。若膜电位累积值超过阈值，则将膜电位复位，即将膜电位复位值(***Vc*** 的值)写入 ***Vd***；并且发放脉冲，即将脉冲值(***Ve*** 的值)置 1。否则，膜电位无需复位，即将膜电位累积值(***Va*** 的值)写入 ***Vd***；且不发放脉冲，即将脉冲值(***Ve*** 的值)置 0。指令执行完毕后，向量 ***Vd*** 的值表示当前的 8 个膜电位累积值，向量 ***Ve*** 的值表示当前的 8 个脉冲生成情况。

4)脉冲批量生成 SIMD 指令(VSELLTW_S4R)

与 SELGT_S4R 类似，如果采用不占用该国产指令系统“0x1F”操作码的设计方案，虽然无法对脉冲生成运算进行专项支持，但可以继续保持指令系统的开放性和灵活性，而且同样能够实现 SNN 神经元模型的脉冲生成批量处理。

新增一条整数向量条件选择指令(可采用浮点复合运算指令格式的寄存器格式)，作为专用的脉冲批量生成 SIMD 指令。

VSELLTW_S4R 的相关定义如下。

①助记符：VSELLTW_S4R。

/*字整数向量小于 0 时选择生成脉冲(包含 4 个操作数)*/

②格式：VSELLTW_S4R ***Va.rw***, ***Vb.rw***, ***Vc.rw***, ***Vd.ww***。

③指令描述：字向量小于“0”时选择生成脉冲。指令的含义说明类似可得，故略。指令执行完毕后，向量 ***Va***(既是源操作寄存器，也是目标操作寄存器)的值表示当前的 8 个膜电位累积值，向量 ***Vd*** 的值表示当前的 8 个脉冲生成情况。

3.2.3 数据存取(空域传送)指令

1)脉冲数据 SIMD 装入指令(VLDD_NC_S)

针对 SNN 网络所需处理的脉冲序列数据量较大的特点，为了提高数据的空域传送效率和提升类脑计算的指令密度，本文参照该国产指令系统 SIMD 存储装入指令的设计原则，提出类脑计算专用的脉冲数 VLDD_NC_S 和脉冲数据 SIMD 存储指令。

采用带功能域的存储器指令格式(相对于不带功能域的存储器指令，带功能域的存储器指令将 16 b 地址偏移缩短为 12 b，从而在指令中增加 4 b 的功能域)，新增 1 条 SIMD 存储装入指令，作为专用的脉冲数据 SIMD 装入指令，如表 6 所示。

Table 6 Spike-Data SIMD Load Instruction VLDD_NC_S

表 6 脉冲数据 SIMD 装入指令 VLDD_NC_S

操作码与功能码 [15:12]	描述	指令格式
0x1C.6	不可缓存装入 8 倍字整数脉冲	带功能域的存储器指令格式

VLDD_NC_S 的相关定义如下。

①助记符：VLDD_NC_S。

/*不可缓存装入 8 倍字整数脉冲*/

② 格式：VLDD_NC_S ***Va.wo***, ***disp.ab***(***Rb.ab***)。

③ 操作：***Vaddr*** ←〈***Rbv*** + SEXT(***disp***)〉；

Va ← (***Vaddr***)[63:0]。

④指令描述：指令中访存地址 ***Vaddr*** 形成机制为 12 b 移量符号扩展到 64 b 加上寄存器 ***Rb*** 的值。VLDD_NC_S 从指定存储器单元取出 256 b 的 8 倍字整数，并写入向量寄存器 ***Va*** 中。该指令不装填也不影响任何核心内部缓存；是否装填核外缓存可以由具体实现决定。

⑤含义说明：该指令用于将脉冲序列从存储器加载到处理器的向量寄存器中，***Va*** 用于存储加载的脉冲序列数据。

2)脉冲数据 SIMD 存储指令(VSTD_NC_S)

类似地，新增 1 条 SIMD 存储装入指令，作为专用的脉冲数据 SIMD 存储指令。

VSTD_NC_S 的相关定义如下。

①助记符：VSTD_NC_S。

/*不可缓存存储 8 倍字整数脉冲*/

②格式：VSTD_NC_S ***Va.wo***, ***disp.ab***(***Rb.ab***)。

③操作：***Vaddr*** ←〈***Rbv*** + SEXT(***disp***)〉；

(***Vaddr***)[255:0] ←***Vav***[255:0]。

④指令描述：指令中访存地址 ***Vaddr*** 形成机制为 12 b 偏移量符号扩展到 64 b 加上寄存器 ***Rb*** 的值。VSTD_NC_S 指令将向量寄存器 ***Va*** 中 256 b 的 8 倍字整数存储到指定存储器单元中。该指令不装填也不影响任何核心内部缓存；是否装填核外缓存可以由具体实现决定。

⑤含义说明：该指令用于将处理器向量寄存器中的脉冲序列存储到存储器中。***Va*** 是源寄存器，其包含待存储的脉冲序列数据。

3.3 支持 DNN 的指令设计示例

相对于面向 SNN 网络的类脑指令，面向 DNN 网络的专用扩展指令已有较为成熟的研究。例如，HiPU 系列芯片基于自定义的 RISC-V 扩展指令集，较好地在指令集层面实现了对 DNN 计算任务的支持。通过借鉴上述做法，在该国产指令系统进行相应的移植和适配，即可完成从 RISC-V 指令集到该国产指令集的转换，从而实现该国产指令系统对 DNN 计算任务的高效处理。

在本节中，通过采用类似于支持 SNN 的类脑指令设计方法，简要列举 2 条支持 DNN 的专用指令作为示例，用于辅助说明该技术路线的可行性，此外不再一一赘述。

1）膜电位单输入乘累加基本指令（SELEQ_SA）

类似于膜电位单输入累积基本指令（SELEQ_S），在进行基于该国产指令系统的面向 DNN 计算的定制化设计时，可以给出相应的 SELEQ_SA。

需要注意，由于 DNN 网络的特征输入是连续的模拟值而非脉冲值，无法采用条件选择的风格进行计算，因此暂不支持适用于 DNN 计算的膜电位双输入乘累加基本指令。

SELEQ_SA 的相关定义如下。

①助记符：SELEQ_SA。

/*整数乘累加*/

②格式：SELEQ_SA *Ra.rl*, *Rb.rl*, *Rc.rl*, *Rd.wl*。

③操作：*Rd* ← *Rav* × *Rbv* + *Rcv*。

④指令描述：*Ra* 与 *Rb* 相乘后再与 *Rc* 相加，最后结果写入 *Rd*。

⑤含义说明：*Ra* 中存放输入特征值，*Rb* 中存放突触权重值，*Rc* 中存放此前的膜电位累积值。指令执行完毕后，*Rd* 的值表示当前的膜电位累积值。

2）膜电位单组输入乘累加 SIMD 指令（VSELLBCW_SA）。

同理，类似于膜电位单组输入累积 SIMD 指令 VSELLBCW_S，在进行基于该国产指令系统的面向 DNN 计算的定制化设计，可以给出相应的 VSELLBCW_SA。

该指令的相关定义如下。

①助记符：VSELLBCW_SA。

/*向量乘累加*/

②格式：VSELLBCW_SA ***Va.rw***, ***Vb.rw***, ***Vc.rw***, ***Vd.ww***。

③操作：***temp*** ← **0**;

FOR(*i*=0; *i*<8; *i*++)

temp[32×*i*+31: 32×*i*]← ***Vav***[32×*i*+31: 32×*i*] × ***Vbv***[32×*i*+31: 32×*i*] + ***Vcv***[32×*i*+31: 32×*i*];

Vd←***temp***;

ENDFOR

④指令描述：***Va***, ***Vb***, ***Vc***, ***Vd*** 均为存放 8 个字整数的向量寄存器，将 ***Va*** 和 ***Vb*** 中 8 个字整数分别相乘后再与 ***Vc*** 中的 8 个字整数分别相加，最后结果写入 ***Vd*** 中对应位置。

⑤含义说明：***Va*** 中存放输入特征值，***Vb*** 中存放突触权重值，***Vc*** 中存放此前的膜电位累积值。指令执行完毕后，***Vd*** 的值表示当前的膜电位累积值。

4 实现类脑计算指令集的核心微结构设计

为探索类脑计算指令集的具体实现方案，本节基于前文所提出的类脑计算指令集，完成对应类脑核心微结构的初步设计。如前所述，在当前主流的类脑处理器架构中，较为常见的是主从异构的处理器形态，包括主控核心和加速核心[15-16]。其中，主控核心负责管理调度，加速核心发挥计算优势，二者紧密配合、有序协同，共同实现类脑智能应用的高效运行。但是，现有类脑处理器的技术方案中，存在 2 点明显不足：

1）安全可控存在风险。现有异构类脑处理器所采用的主控核心，多数都采用 RISC-V 或 ARM 等核心，国产处理器核心较为少见，因此存在潜在的安全威胁和自主可控风险。

2）信息交互代价较大。主控核心和加速核心相互独立，二者之间存在频繁的信息交互，需要进行专门的协议约束和接口设计，并且传输延迟和通信开销较大。

立足于实现“自主可控”和“安全可靠”的迫切需求，并着眼于主控核心和加速核心之间的信息交互代价较大的问题，本文提出了一种基于某国产处理

器的类脑核心(称为“类脑增强核心”)微结构设计方案。

该设计不再采用主从异构的类脑处理器形态，而是在该国产处理器核心的内部细粒度地集成一个能够兼容 SNN 和 DNN 计算的类脑处理部件，并对指令和数据通路上的相关部件进行类脑计算配套功能的支持和完善。在保持通用计算处理能力的同时，兼容 DNN/SNN 的神经网络计算任务，实现了类脑计算的专项功能增强。通过这种方式，既能够实现完全的自主可控，又避免了核心对外的频繁信息交互，从而在根本上给出了以上 2 个难题的解决途径。

为了尽最大可能地保持兼容性，本文所提出的类脑增强核心，保留了原国产处理器核心的所有关键部件，因而继承了其全部的通用计算功能。相对于该国产处理器核心，类脑增强核心的主要变化在于：新增了 1 个类脑处理部件(BIP)，负责类脑相关的运算和访存双重任务；同时为了适应这一变化，对指令流水线上涉及的其他部件做出相应的修改。类脑增强核心的微结构如图 1 所示。

4.1 类脑增强核心的指令前段部件

指令前段部件(instruction fetch/prediction/decode, IFPD)从功能上可以划分为 3 个部分：取指控制、转移预测以及译码整理。其主要任务是根据当前取指地址，读取指令信息并进行整理分析，完成指令的动态转移预测，进行指令的分组打包和译码，然后送至指令后段部件进行处理。

为了支持类脑计算的专用指令集，需要对 IFPD 部件的译码整理模块(尤其是指令译码)进行功能强化，形成类脑计算指令执行所需的功能部件号指示信息、控制信息和数据信息，生成源寄存器和目标寄存器，包括有效位、类型和逻辑寄存器号，同时对系统寄存器进行逻辑更名，从而支持对类脑指令的正确识别和正确译码。

4.2 类脑增强核心的指令后段部件

指令后段部件(instruction rename/dispatch/commit, IRDC)的主要功能为：指令重命名、发射和提交，以及重排序。IRDC 部件接收从 IFPD 发过来的指令，完成重命名后，根据指令发射端口为其分配发射队列。发射队列可进一步划分为多个子队列，分别用于存放发往不同流水线的指令。

为了实现对 DNN/SNN 神经网络计算任务的高效支持，在类脑增强核心中，为类脑专用指令设置独立的发射队列 NNet，存放发往 BIP 的执行流水线的运算类指令，以及发往 BIP 部件的访存流水线的

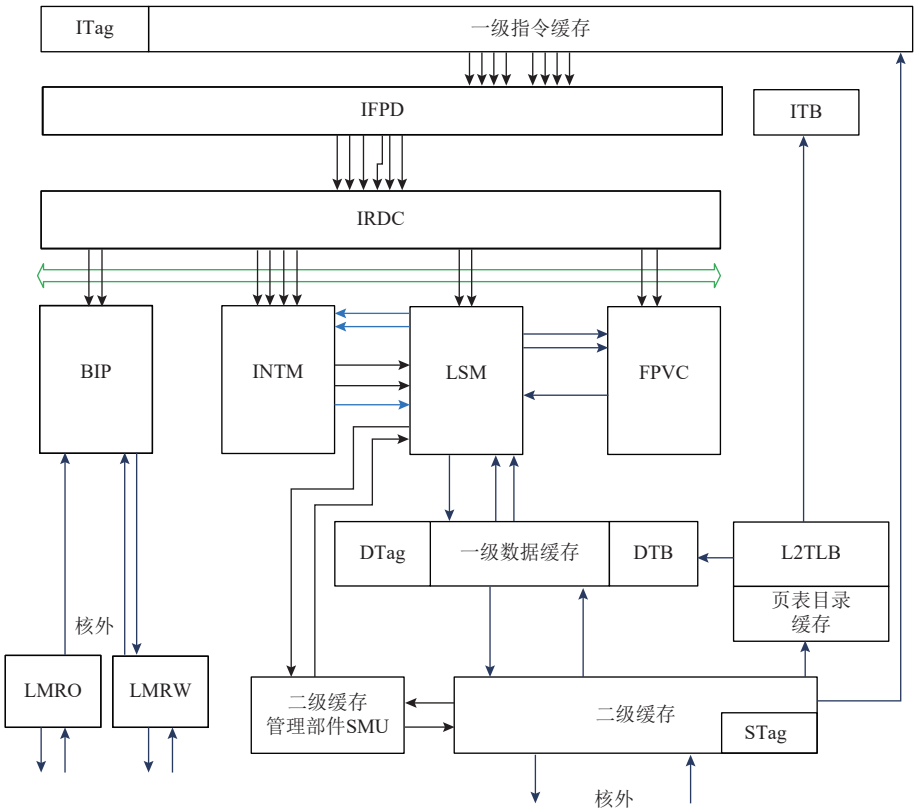


Fig. 1 Micro-architecture of neuromorphic enhanced core embedded with DNN/SNN computation

图 1 兼容 DNN/SNN 计算的类脑增强核心微结构

Load/Store 类指令。

相对于原国产处理器核心，类脑增强核心每拍额外增加最多 2 条指令进入 NNet 发射队列。对于非类脑专用指令，分别进入 2 个子队列中，其入队逻辑完全一致；对于类脑专用指令，单独进入 NNet 队列，进行独立管理。

4.3 类脑增强核心的类脑处理部件

类脑增强核心新增一个类脑处理部件（brain-inspired processing, BIP），与其他部件共用整数和浮点的寄存器文件资源，但设置独立的 2 个类脑指令发射端口、1 条类脑运算流水线、1 条类脑访存流水线，以及专用控制模块等，其主要功能分为运算和访存 2 个方面。

1）在运算方面，其主要功能为：接收指令发射站台的类脑专用指令执行请求；在读寄存器站台，从共用寄存器文件中读取源操作数；在指令执行站台，按指令功能执行整数、浮点或 SIMD 运算操作，得出运算结果；在指令完成站台，将结果写入寄存器文件，并向重排序缓冲报告指令执行的完成情况。

2）在访存方面，其主要功能为：接收来自指令后段部件 IRDC 产生的访存请求，控制访存请求对本地存储 LMRO（local memory read-only）和 LMRW（local memory read-write）的读写访问，控制类脑数据存取指令的乱序执行。

从 BIP 的主要功能可以看出，BIP 作为类脑增强核心的类脑指令专属处理部件，承担了类似于整数处理部件（INTM）、浮点向量部件（FPVC）和访存处理部件（LSM）三者对于非类脑专用指令的处理功能。BIP 部件与其他部件协作配合，通过寄存器文件进行必要的信息交互，共同完成对 DNN/SNN 神经网络计算任务的专项支持和加速处理。

4.4 类脑增强核心的本地存储 LMRO 和 LMRW

基于类脑增强核心的类脑处理器层次化存储体系结构主要由 3 级存储层次组成，即寄存器、片上存储空间（类脑增强核心的 2 级缓存和片上 SRAM 等）以及内存空间（DDR 存储）。

由于卷积相关的数据通常不具备时间局部性，因此不适于采用缓存管理机制。为高效支持类脑增强核心内部 BIP 部件的运算，本文采用固定地址的方式，在片上存储空间采用了能够大幅加快处理速度的局部存储设计：用于存放特征数据或脉冲阵列缓存的 LMRW，以及用于存放突触权重缓存的 LMRO。

从外部 DDR 存储器中，将神经网络的输入特征数据或脉冲阵列搬运至 LMRW；神经网络的突触权重，存放于片上的 LMRO。根据 NNet 指令队列中的类脑专用指令，由控制模块生成具体的控制信号，并控制神经元阵列进行数据读取，然后启动后续计算过程。

5 本文类脑计算指令的特点分析

5.1 基于某国产指令系统的类脑指令列表

通过挖掘 SNN 网络的计算特点，凝练与特定计算操作相适配的类脑算子，本文一共提出了 12 条基于某国产指令系统的类脑计算指令。其中，10 条为面向 SNN 网络的专用指令，2 条为面向 DNN 网络的专用指令。在保证与该国产指令系统向前兼容的前提下，将本文所提出的类脑计算指令按操作码排序，如表 7 所示。

5.2 主要设计特点和未来研发方向

5.2.1 主要设计特点

本文所提出的基于某国产指令系统的类脑指令集设计，为该国产处理器的类脑计算能力的拓展和赋能奠定了基础，为实现自主可控的类脑智能处理器研发提供了关键技术支撑。

基于该国产指令系统的类脑指令集的主要设计特点，可以概括为 3 点：

1）既能够完成复杂的通用计算和管理调度等任务，也实现了对 DNN 计算和 SNN 计算的特定支持和高效兼容。

2）类脑计算专用指令的设计符合该国产指令系统的既有设计原则，保持了对现有指令集版本的向前兼容。

3）提出了多种类型的膜电位累积指令和脉冲生成指令，并提出了 2 种与现有国产指令系统兼容的可行方案（操作码占用与否），便于根据不同的网络规模、领域任务等特定应用情况，有针对性地选取具体实现方式。

5.2.2 未来研发方向

在未来的研发过程中，如果相关条件具备，将进一步致力于在新一代国产处理器核心和芯片上进行逻辑实现，通过编译器和操作系统的配套优化，完成仿真实验分析和初步评估，证实该类脑指令系统功能正确，并针对典型应用场景进行功能验证和性能测试，逐步完成针对处理器实际效能的验证测试和综合评估。

基于该国产指令系统的类脑指令集设计的未来

Table 7 A List of the Neuromorphic Instructions Based on A Domestic Instruction System (in Order of the Opcode)

表 7 基于某国产指令系统的类脑指令列表（按操作码排序）

序号	用途	操作码	功能码	指令	类型	指令说明	指令格式
1	膜电位累积（时域积分）	0x11	[12:10]=0x0 [15:13]=0x1	SELEQ_S	膜电位单输入累积基本指令	整数等于“0”时选择累积	整数复合运算指令格式
2		0x11	[12:10]=0x0 [15:13]=0x2	SELEQ_D	膜电位双输入累积基本指令	2 个整数等于“0”时分别选择累积	整数复合运算指令格式
3	脉冲生成（时域发放）	0x11	[12:10]=0x0 [15:13]=0x3	SELGT_S4R	脉冲生成基本指令	整数大于“0”时选择生成脉冲	整数复合运算指令格式
4	膜电位累积（时域积分）	0x11	[12:10]=0x0 [15:13]=0x4	SELEQ_SA	膜电位单输入乘累加基本指令	整数乘累加（支持 DNN）	整数复合运算指令格式
5		0x1B	[15:10]=0x16	VSELLBCW_SA	膜电位单输入乘累加 SIMD 指令	向量乘累加（支持 DNN）	浮点复合运算指令格式
6	脉冲生成（时域发放）	0x1B	[15:10]=0x17	VSELLTW_S4R	脉冲批量生成 SIMD 指令	字向量小于“0”时选择生成脉冲	浮点复合运算指令格式
7	膜电位累积（时域积分）	0x1B	[15:10]=0x1C	VSELLBCW_S	膜电位单组输入累积 SIMD 指令	字向量低位为“0”时选择累积	浮点复合运算指令格式
8		0x1B	[15:10]=0x1D	VSELLBCW_D	膜电位双组输入累积 SIMD 指令	字向量低 2 位为“0”时分别选择累积	浮点复合运算指令格式
9	数据存取（空域传送）	0x1C	[15:12]=0x6	VLDD_NC_S	脉冲数据 SIMD 装入指令	不可缓存装入 8 倍字整数脉冲	带功能域的存储器指令格式
10		0x1C	[15:12]=0x7	VSTD_NC_S	脉冲数据 SIMD 存储指令	不可缓存存储 8 倍字整数脉冲	带功能域的存储器指令格式
11	脉冲生成（时域发放）	0x1F	[15]=0x0	SELGT_S5R	脉冲生成基本指令	整数大于“0”时选择生成脉冲	类脑运算指令格式
12		0x1F	[15]=0x1	VSELLTW_S5R	脉冲批量生成 SIMD 指令	字向量小于“0”时选择生成脉冲	类脑 SIMD 运算指令格式

研发方向，可以概括为 6 点：

1）参考模型构建。针对类脑指令集，建立功能细节完善的指令集参考模型模拟器，实现类脑指令系统的功能仿真。

2）模拟环境验证。为参考模型构建处理器的模拟验证环境，提升模拟器运行速度，为开展正确性验证提供基础依据。

3）仿真实验分析。基于指令集参考模型和模拟验证环境，对不同的类脑应用程序进行实验测试，分析测试程序的反汇编代码，确认类脑指令的功能正确性。同时，在基于通用指令集架构的处理器上运行相同的测试程序，并将二者的反汇编结果进行对比，分析指令序列调用和指令密度情况的差别，从而对类脑指令集支持下的处理能力提升进行实验评估。

4）学习规则支持。目前所提出的类脑指令集设计主要聚焦于推理能力，为使类脑处理器具备片上学习和训练能力，需要针对常见的学习规则，在指令集层面进行专门的设计支撑，从而为未来的类脑边缘大模型的训练部署等潜在需求，做好相应的功能性支持和拓展。

5）复杂向量扩展。向量化扩展已经成为主流处理器指令集的发展方向（例如 RISC-V 和 ARMv9 均提出了不同长度的复杂向量扩展指令集），未来需要在该国产处理器的类脑指令集上进行更为复杂的向量化扩展和实现。

6）软件生态构建。由于基于自主指令构建，软件生态相对薄弱，因此该国产处理器系统当前还存在实际程序运行性能和用户体验有待提高等问题。因此，在类脑智能处理器的发展初期，就应重视基础软件与底层硬件之间的协同设计，更好地发挥自主软硬件协同设计的优势和潜力，最大化提高系统的可扩展性、兼容性和通用性。

6 总 结

端边云协同计算系统是分布式计算的典型体现之一，为了真正实现端边云协同计算系统的融合优势，不仅需要系统级和网络级的顶层设计优化，也需要节点级和硬件级的底层设计创新。指令系统是软硬件的交互界面，是处理器架构设计的重点之一。指令系统的设计不仅要全面、完备、高效、合理，还需要结合新兴应用的不断发展，对指令集进行适应性的迭代优化和丰富拓展。针对分布式系统、端边云协同和类脑计算等新兴领域的特定应用需求，本文提出了一种适用于分布式系统中的终端硬件设备的指令集，能够在通用软件支撑环境和异构硬件平台之间搭建起一座桥梁，从而降低分布式计算系统的部署和升级难度，高效推进端边云系统的协同演进。具体地，本文提出了 12 条基于某国产指令系统的类脑计算指令，完成了类脑增强核心的微结构设计，为国产处理器系统的长远发展提供了有力支撑，并为类脑计算赋能分布式计算系统奠定了技术基础。

作者贡献声明：冯烁、路冬冬、尹飞负责主体设计工作并撰写论文；杨剑新、班冬松、何军负责设计思路指导和改进建议；颜世云、李媛、睢浩宇负责论文修改和完善工作。

参 考 文 献

[1] Ng J S, Lim W Y B, Luong N C, et al. A comprehensive survey on coded distributed computing: Fundamentals, challenges, and networking applications[J]. IEEE Communications Surveys and Tutorials, 2021, 23(3): 1800−1837

[2] Wu Qihui, Ding Guoru, Xu Yuhua, et al. Cognitive Internet of things: A new paradigm beyond connection[J]. IEEE Internet of Things Journal, 2014, 1(2): 129−143

[3] Li Guoqi, Deng Lei, Tang Huajing, et al. Brain inspired computing: A systematic survey and future trends[J]. Proceedings of the IEEE, 2024, 112(6): 544−584

[4] Schuman C D, Kulkarni S R, Parsa M, et al. Opportunities for neuromorphic computing algorithms and applications[J]. Nature Computational Science, 2022, 2(1): 10−19

[5] Mead C, Neuromorphic electronic systems[J]. Proceedings of the IEEE, 1990, 78(10): 1629−1636

[6] Roy K, Jaiswal A, Panda P. Towards spike-based machine intelligence with neuromorphic computing[J]. Nature, 2019, 575(7784): 607−617

[7] Yang Zhijie, Wang Lei, Shi Wei, et al. Asynchronous network-on-chip architecture for neuromorphic processor[J]. Journal of Computer Research and Development, 2023, 60(1): 17−29 (in Chinese)
(杨智杰, 王蕾, 石伟, 等. 类脑处理器异步片上网络架构[J]. 计算机研究与发展, 2023, 60(1): 17−29)

[8] Qu Peng, Chen Jiajie, Zhang Youhui, et al. A proposal of software-hardware decoupling hardware design method for brain-inspired computing[J]. Journal of Computer Research and Development, 2021, 58(6): 1146−1154 (in Chinese)
(渠鹏, 陈嘉杰, 张悠慧, 等. 实现软硬件解耦合的类脑计算硬件设计方法[J]. 计算机研究与发展, 2021, 58(6): 1146−1154)

[9] Zhao Wenzhe, Yang Guoming, Xia Tian, et al. HIPU: A hybrid intelligence processing unit with fine-grained ISA for real-time deep neural network inference applications[J]. IEEE Transactions on Very Large Scale Integration Systems, 2023, 31(12): 1980−1993

[10] Shen Juncheng, Ma De, Gu Zonghua, et al. Darwin: A neuromorphic hardware co-processor based on spiking neural networks[J]. Science China Information Sciences, 2016, 59(2): 1−5

[11] Ma De, Jin Xiaofei, Sun Shichun, et al. Darwin3: A large-scale neuromorphic chip with a novel ISA and on-chip learning[J], National Science Review, 2024, 11(5): 1−17

[12] Painkras E, Plana L A, Garside J, et al. SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation[J]. IEEE Journal of Solid-State Circuits, 2013, 48(8): 1943−1953

[13] Hoppner S, Yan Yexin, Dixius A, et al. The SpiNNaker 2 processing element architecture for hybrid digital neuromorphic computing[J]. arXiv preprint, arXiv: 2103.08392, 2021

[14] Intel. Taking neuromorphic computing to the next level with Loihi 2[R/OL]. [2025-04-09]. https://download.intel.com/newsroom/2021/new-technologies/neuromorphic-computing-loihi-2-brief.pdf

[15] Pei Jing, Deng Lei, Song Sen, et al. Towards artificial general intelligence with hybrid TianJic chip architecture[J]. Nature, 2019, 572(7767): 106−111

[16] Deng Lei, Wang Guanrui, Li Guoqi, et al. Tianjic: A unified and scalable chip bridging spike-based and continuous neural computation[J]. IEEE Journal of Solid-State Circuits, 2020, 55(8): 2228−2246



Feng Shuo, born in 1989. PhD, engineer. His main research interests include general-purpose processor architecture, micro-architecture design, and brain-inspired computing.

冯　烁, 1989 年生。博士, 工程师。主要研究方向为通用处理器架构、核心微结构设计、类脑智能计算。



Lu Dongdong, born in 1988. Master, senior engineer. His main research interests include general-purpose processor architecture, micro-architecture design, and ISA design.

路冬冬, 1988 年生。硕士, 高级工程师。主要研究方向为通用处理器架构、核心微结构设计、指令集架构设计。



Yin Fei, born in 1978. Master, senior engineer. Her main research interests include high performance computer architecture, multi-core processor architecture, and ISA design.

尹　飞, 1978 年生。硕士, 高级工程师。主要研究方向为高性能计算机体系结构、多核处理器架构、指令集架构设计。



Yang Jianxin, born in 1976. Master, senior engineer. Senior member of CCF. His main research interests include computer architecture and high performance processor design.

杨剑新, 1976 年生。硕士, 高级工程师。CCF 高级会员。主要研究方向为计算机体系结构、高性能处理器设计。



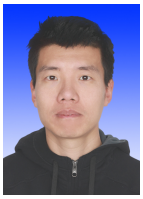
Ban Dongsong, born in 1982. PhD, senior engineer. His main research interests include high performance processor design, SoC design, and brain-inspired computing.

班冬松, 1982 年生。博士, 高级工程师。主要研究方向为高性能处理器设计、片上系统设计、类脑智能计算。



He Jun, born in 1980. PhD, senior engineer. His main research interests include computer architecture and high performance general-purpose processor design.

何　军, 1980 年生。博士，高级工程师。主要研究方向为计算机体系结构、高性能通用处理器设计。



Yan Shiyun, born in 1981. Master, senior engineer. His main research interests include high performance computer architecture, VLSI design & verification.

颜世云, 1981 年生。硕士，高级工程师。主要研究方向为高性能计算机体系结构、超大规模集成电路设计与验证。



Li Yuan, born in 1976. Master, senior engineer. Her main research interests include computer architecture and high performance processor design.

李　媛, 1976 年生。硕士，高级工程师。主要研究方向为计算机体系结构、高性能处理器设计。



Ju Haoyu, born in 2000. Bachelor, engineer. His main research interest includes VLSI design & verification.

睢浩宇, 2000 年生。学士，工程师。主要研究方向为超大规模集成电路设计与验证。