体系结构模拟器的研究现状、挑战与展望

张锦[1]，陈铸[1]，陈照云[2,3*]，时洋[2,3]，陈冠军[2,3]

（1.长沙理工大学计算机与通信工程学院，湖南 长沙 410000；
2.国防科技大学计算机学院，湖南 长沙 410000；3.先进微处理器芯片与系统重点实验室，湖南 长沙 410000）

摘　要：在众多科学领域的研究与开发中，模拟器都扮演着不可替代的角色。在体系结构领域尤其如此，模拟器提供了一个安全、成本低廉的虚拟环境，使研究人员能够快速开展实验分析和评测。同时，模拟器还可以加速芯片设计和验证的过程，从而节省时间和资源成本。然而，随着处理器体系结构的演化进步，尤其是专用处理器发展呈现多元化特点，为了能够对体系结构设计探索提供重要的反馈，模拟器的重要作用日益凸显。综述了体系结构模拟器目前的发展与应用现状，重点介绍了几种目前较为典型的体系结构模拟器。通过对专用于不同处理器的模拟器技术手段的分析，深入了解不同架构下模拟器的侧重点及技术难点。此外，还对体系结构模拟器未来发展的关键点进行了思考与评述，以展望其在处理器设计研究领域的前景。

关键词：模拟器；体系结构；处理器；芯片设计反馈；虚拟化

中图分类号：TP302.1　　**文献标志码：**A　　**DOI：**10.19678/j.issn.1000-3428.0068870

Current Research Status, Challenges, and Future Prospects of Architectural Simulators

ZHANG Jin[1], CHEN Zhu[1], CHEN Zhaoyun[2,3*], SHI Yang[2,3], CHEN Guanjun[2,3]

(1. School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410000, Hunan, China;
2. College of Computer, National University of Defense Technology, Changsha 410000, Hunan, China;
3. Key Laboratory of Advanced Microprocessor Chips and Systems, Changsha 410000, Hunan, China)

【Abstract】 Simulators play an indispensable role in an array of scientific fields involving research and development. Particularly in architectural design, simulators provide a secure and cost-effective virtual environment, enabling researchers to conduct rapid experimental analyses and evaluations. Simultaneously, simulators facilitate the acceleration of the chip design and verification processes, thereby conserving time and reducing resource expenditure. However, with the evolutionary advances in processor architectural designs—specifically, the flourishing diversifications featured in dedicated processors—the key role played by simulators in providing substantial feedback for architectural design exploration has gained prominence. This discourse provides an overview of the current developments and applications of architectural simulators, accentuating a few illustrative examples. Analyzing the techniques employed by simulators dedicated to various processors allows for a deeper understanding of the focal points and technical complexities under different architectures. Moreover, this discourse deliberates speculative assessments and critiques of vital aspects of future architectural simulator developments, aspiring to forecast their prospects in the field of processor design research.

【Key words】 simulator; architecture; processor; chip design feedback; virtualization

0　引言

计算机体系结构模拟器这一概念最早出现于计算机发展的早期阶段，大约在 20 世纪 50 年代到 60 年代。由于当时计算机硬件稀缺且昂贵，研究人员开始采用模拟技术来运行计算机程序并进行算法调试。随着计算机技术的发展，模拟器也随之演化，始终与计算机技术的发展紧密相连。从 20 世纪 70 年代到 80 年代，随着计算机体系结构的进步和性能的提升，大规模的系统级模拟器开始涌现。这些模拟器能够模拟整个计算机系统，包括处理器、内存和设备等。到了 20 世纪 90 年代，个人计算机和工作站的普及使得模拟器变得更加普遍和易用，研究人员对模拟器的需求也越来越广泛。这个时期出现了各种多样化的模拟器，包括指令集模拟器[如 SPIM(Simulator for MIPS Processors)[1]]和微体系结构模拟器[如 RSIM(Rice Simulator for ILP Multiprocessors)]。这些模拟器可用于仿真和评估

收稿日期：2023-11-20　　**修回日期：**2024-03-13

基金项目：湖南省教育厅优秀青年项目(22B0341)。

通信作者 E-mail：* chenzhaoyun@nudt.edu.cn

计算机体系结构、开发编译器、调试代码等。自21世纪初以来，随着并行计算和新兴领域(如网络和图形处理)的快速发展，专用于这些领域的模拟器逐渐涌现。例如，图形处理器(GPU)模拟器[如GPGPU-Sim(General-Purpose Graphics Processing Unit Simulator)[2]]用于模拟并行计算和GPU的行为，网络模拟器[如NS-3(Network Simulator 3)[3]]用于建模和仿真计算机网络。如今，在人工智能和深度学习的兴起中，神经网络处理器模拟器和数字信号处理器(DSP)模拟器等新型的专用模拟器也逐渐涌现。

现代体系结构模拟器提供了高度真实的仿真能力，为研究人员和开发人员提供了强大的工具，支持算法优化、性能评估和系统调试等任务[4]。如今，体系结构模拟器已具备如下强大功能：

1)虚拟机：体系结构模拟器可以创建虚拟机，在同一台物理计算机上运行多个操作系统，例如Windows、Linux、macOS等，每个操作系统都像独立的计算机一样工作。

2)应用程序兼容性测试：体系结构模拟器可以用于测试应用程序在不同操作系统或不同硬件平台上的兼容性。用户可以在模拟器中运行应用程序，并评估其在不同环境下的表现。

3)硬件模拟：一些体系结构模拟器可以模拟特定硬件设备的功能，例如手机、平板电脑、游戏主机等。这使得开发人员可以在模拟环境中测试和调试软件，而无需实际的硬件设备。

4)软件开发和调试：体系结构模拟器为开发人员提供了一个安全且可控的环境，用于开发、调试和测试软件。它们可以帮助开发人员捕获和修复应用程序中的错误和问题。

5)体系结构设计反馈：体系结构模拟器能够帮助用户评估和优化程序性能，分析代码执行路径和循环迭代次数，提供内存布局和优化的反馈，以及评估不同硬件配置和算法在能耗方面的差异。

这些功能强大的体系结构模拟器已经成为计算机领域重要的辅助工具之一，推动了计算机科学的进步和创新。

体系结构模拟器发展至今，已经形成了一个庞大的家族。根据模拟的级别、应用领域和技术等分类方式[5-6]，体系结构模拟器可以分为多种类别。本文将主要介绍几种应用于典型通用处理器(CPU)、专用处理器[7]领域的模拟器，从CPU、GPU、DSP、神经网络处理器模拟器出发，以QEMU(Quick Emulator)[8]、Gem5(Generalized Execution-driven Multiprocessor Simulator 5)[9]、GPGPU-Sim等具有代表性的模拟器为例，分析不同架构下模拟器的侧重点与技术难点，对计算机体系结构模拟器研究现状和未来的发展进行总结与展望。

1 计算机体系结构模拟器简介

1.1 体系结构模拟器的分类

模拟器的分类方式有许多种，根据模拟的目标组件、精度、输入和推进方式的不同，可以将模拟器进行不同的分类。本节主要介绍模拟级别和处理器类型这两种常见的分类[10]。这两种分类标准符合计算机科研领域的划分标准，并且对于大部分从事与计算机相关科研工作的人员来说具有参考价值。除了上述两种以外，还可以根据工作原理与应用领域进行分类，如图1所示(彩色效果见《计算机工程》官网HTML版，下同)。

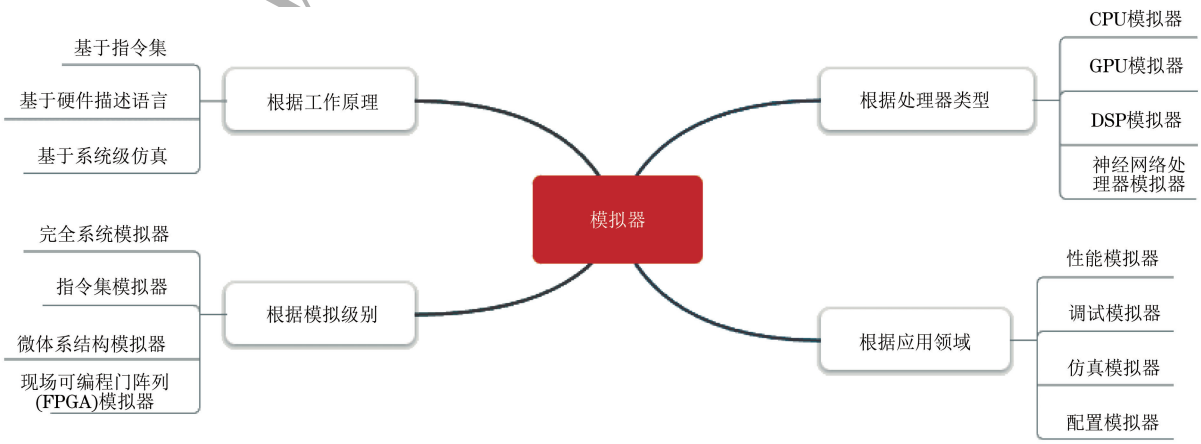

图1 体系结构模拟器分类

Fig.1 Classification of architectural simulators

1.1.1 按模拟级别的分类

1)完全系统模拟器：这种模拟器模拟整个计算机系统，包括处理器、内存、外设等。它能够运行目标体系结构的操作系统和应用程序，并提供对应的

指令集和硬件特性。完全系统模拟器通常用于软件开发、调试和验证。例如,QEMU、Bochs[11]等。

2)指令集模拟器:这种模拟器模拟处理器的指令集架构和行为,不涉及具体的物理硬件。它能够解释和执行目标体系结构的指令,实现对应的运算和功能。指令集模拟器常被用于编译器开发、代码优化和体系结构研究。例如,SPIM。

3)微体系结构模拟器:这种模拟器模拟处理器的微体系结构层面,考虑了更详细的硬件细节和性能特性。它能够模拟处理器的流水线、缓存、分支预测等微体系结构组件,具有详尽的性能分析和仿真能力。微体系结构模拟器常被用于体系结构评估、优化和性能调优。例如,Gem5-gpu[12]、PinPlay[13]等。

4)FPGA 模拟器:这种模拟器使用 FPGA 来模拟处理器的行为。它将处理器的寄存器传输级(RTL)描述与 FPGA 技术相结合,实现对处理器的逐周期模拟。FPGA 模拟器常被用于验证新的处理器设计、验证和调试硬件。例如,RAMP(Research Accelerator for Multiple Processors) gold[14]。

1.1.2 按处理器类型的分类

1)CPU 模拟器:这种模拟器模拟 CPU 架构,例如 x86、高级精简指令集机器(ARM)、MIPS 等。这种模拟器广泛应用于软件开发、编译器研究和系统级仿真等领域。它允许开发人员在不同的处理器架构上运行和测试程序,以确保其在不同平台上的正确性和性能。例如,QEMU、Gem5 等。

2)GPU 模拟器:这种模拟器是一种用于模拟和调试 GPU 的行为和性能的工具。它被广泛用于图形编程、游戏开发、计算机视觉和科学计算等领域。GPU 模拟器能够模拟 GPU 的并行计算单元、内存层次结构、像素渲染管线等关键组件,以便开发人员可以在开发和优化 GPU 相关的应用程序时进行实验和调试。例如,GPGPU-Sim、Barra[15]等。

3)DSP 模拟器:这种模拟器是用于模拟和评估 DSP 行为和性能的工具。它被广泛用于数字信号处理算法的开发、优化和调试。DSP 模拟器能够模拟 DSP 的计算单元、并行架构、存储器层次结构和指令执行等关键组件,以便开发人员可以在开发和优化 DSP 相关应用程序时进行实验和调试。例如,德州仪器(TI) CCS(Code Composer Studio)[16]、CEVA-Toolbox[17]等。

4)神经网络处理器模拟器:这种模拟器是一种用于模拟和评估深度学习任务的硬件加速器的工具。它们被广泛用于深度学习算法的开发、优化和调试。神经网络处理器模拟器能够模拟硬件加速器的计算单元、内存架构、数据流和指令执行等关键组件,以便开发人员可以在开发和优化深度学习应用程序时进行实验和调试。例如,MAESTRO[18]、SCALE-Sim[19]等。

体系结构模拟器的多样性使得研究人员能够在虚拟环境中进行各种实验,无论其研究的对象是硬件还是软件。通过体系结构模拟器,研究人员可以模拟和观察不同的场景和情况,以评估性能、验证设计、测试假设等。这种能力使得体系结构模拟器成为科研工作中的重要工具,无论是在计算机体系结构、芯片设计、操作系统开发、编译器优化,还是其他相关领域。研究人员可以根据自己的需求选择合适的模拟器,以进行实验和验证,推动科学研究的进展。

1.2 体系结构模拟器的常用技术

体系结构模拟器使用多种技术来实现模拟其他系统或设备的行为和功能。以下是一些常用的技术介绍:

1)虚拟化技术:虚拟化技术是体系结构模拟器的核心技术之一。它允许在物理计算机上创建多个虚拟的计算环境,每个环境都可以独立运行操作系统和应用程序。常见的虚拟化技术包括全虚拟化和半虚拟化技术。

2)模拟器引擎:模拟器引擎是体系结构模拟器的主要组成部分,用于执行指令集和处理模拟环境中的各种操作。它可以模拟处理器、内存、设备等硬件组件,以便执行模拟操作。

3)反汇编和指令解释:在模拟过程中,体系结构模拟器需要将目标系统的原始机器码转换为可执行指令。为此,体系结构模拟器首先使用反汇编技术将机器码反向转换为可读的汇编语言,然后使用指令解释器将其转换为可执行指令。例如,动态二进制翻译(DBT)技术、JIT(Just-In-Time)翻译技术。

4)硬件抽象层:为了实现硬件设备的模拟,体系结构模拟器通常使用硬件抽象层。这是一个接口层,它将体系结构模拟器和物理设备之间的交互抽象出来,使得体系结构模拟器可以模拟各种硬件设备。例如,输入输出设备、网络接口等。

5)仿真器和调试器:一些体系结构模拟器还配备了仿真器和调试器,用于模拟执行目标系统和调试应用程序。这些工具可以帮助开发人员追踪和诊断代码问题,以及模拟系统的运行情况。

通过这些技术,体系结构模拟器能够提供准确的模拟环境,使科研人员能够在计算机上运行

和测试各种架构下的操作系统、应用程序或硬件设备。

2 CPU 模拟器

CPU 作为最早出现也是最为常用的处理器，应用于它的模拟器也是当前数量最多、性能最强的。如今，人们常用的有 QEMU、Gem5、Bochs 等。截至 2023 年 9 月，在 EI 数据库中以 QEMU 与 Gem5 为关键词检索的相关论文数量分别为 683 与 550 篇，是体系结构模拟器相关论文中占比最多的两种，所以本节将以 QEMU 与 Gem5 为例，分析 CPU 模拟器目前的发展状况。

2.1 QEMU 模拟器

QEMU 最早由 Fabrice Bellard 于 2003 年开发。当时，虚拟化技术尚不成熟，市场上的虚拟机软件大多是商业闭源的，缺乏灵活性和可定制性。Fabrice Bellard 希望开发一款可自由定制的虚拟机监视器和模拟器，以满足不同用户的需求。他创建了 QEMU 项目，并发布了开源代码，这使得 QEMU 能够吸引到全球的开发者和用户的参与。在开源社区的共同努力下，QEMU 逐渐成为一款功能强大、跨平台的虚拟化工具。它不仅支持多种硬件架构和操作系统，而且还具备了硬件仿真以及灵活的网络和磁盘管理等特点。

随着时间的推移，QEMU 不断得到改进和增强，提供了更多的功能和性能优化。它已经成为开源虚拟化领域中备受推崇和广泛使用的工具之一。其主要特点如下：

1）虚拟化：QEMU 可以用作虚拟化平台，支持在不同的体系结构上模拟和运行各种操作系统。

2）体系结构支持：QEMU 支持模拟多种体系结构，包括 x86、ARM、MIPS、PowerPC 等，各种操作系统可在这些体系结构上运行。

3）翻译技术：QEMU 主要使用 DBT 技术。TCG（Tiny Code Generator）是 QEMU 实现 DBT 的关键，它将这些原始指令动态地翻译成中间表示（IR），再将其转换为宿主平台的指令集架构。这种 IR 的转换和优化过程是 DBT 的核心。

4）设备模拟：QEMU 提供丰富的设备模拟，包括网络设备、存储设备、显卡和声卡等，使得模拟环境更加完整。

2.2 Gem5 模拟器

Gem5 最初是由美国密歇根大学和加州大学伯克利分校的研究团队合作开发的，是一个基于事件驱动的多组件系统模拟器，结合了密歇根大学的 M5[20] 项目和加州大学伯克利分校的 GEMS[21] 项目。从那时起，Gem5 开始了其独立的开源奇幻之旅，并得到了来自全球众多研究机构和公司的支持。

随着时间的推移，Gem5 不断发展壮大，逐渐成为理论研究、教育培训和实际应用的重要工具。Gem5 的开发团队不断致力于提升其性能、可扩展性和可配置性，并不断引入新的功能和特性。Gem5 还积极参与学术会议和研讨会，使其成为计算机体系结构研究领域的重要组成部分。

总体而言，Gem5 从最初的原型发展到如今的开源项目，通过不断的改进，已成为一款领先的计算机体系结构模拟器，为学术界和工业界提供了强大的工具和平台。未来，Gem5 将继续发展壮大，为计算机体系结构领域的研究和开发做出更大的贡献。其主要特点如下：

1）多组件模拟：Gem5 可以模拟整个计算机系统的多个组件，包括 CPU、内存层次结构、硬盘、缓存、网络等。

2）多核支持：Gem5 适用于多核系统的模拟与研究，支持多种 CPU 模型和内存一致性协议。

3）翻译技术：Gem5 使用了多种翻译技术，包括静态编译和 JIT 翻译，用于将目标体系结构的指令翻译为宿主机上的本地指令。

4）CPU 多模型：Gem5 包括 4 种不同的 CPU 模型，即 AtomicSimple、In-Order、Out-of-Order（简称 O3）和 TimingSimple，每种模型的具体实现和 ISA（Instruction Set Architecture）不关联，因此 CPU 模型和 ISA 间可以任意搭配以组成多种计算机系统。

2.3 QEMU 与 Gem5 的对比

QEMU 与 Gem5 作为目前两种主流的 CPU 模拟器都有着各自的用武之地。QEMU 的设计目标是提供一个高性能的虚拟机和模拟器平台，能够在不同的处理器体系结构上运行不同的操作系统，主要关注虚拟化和可移植性。Gem5 的设计目标是提供一个通用、可扩展的多处理器系统模拟器，主要关注对多核系统、内存一致性和硬件模型的支持，能够进行更细粒度的性能评估和体系结构研究。QEMU 在虚拟化、交叉编译、系统仿真和调试等领域得到广泛应用，可以用于开发、测试和部署各种软件和系统。Gem5 主要应用于计算机体系结构、操作系统和编译器优化等领域的研究和教育，为研究人员和学生提供了一个灵活的模拟环境，用于体系结构评估、性能优化和系统设计验证等任务。

两种模拟器也有着许多相似之处，例如在多平台、虚拟化、硬件模拟等方面。在运行方式上，QEMU 有两种运行模式，分别是系统态与用户态，用户态模式仅模拟应用程序的运行环境，而系统态模式模拟整个操作系统和硬件环境，如图 2 所示。在 Gem5 上同样有着类似的两种模式，称为全系统模式与系统调用模式，分别对应 QEMU 的系统态与用户态。其运行模式接近于 QEMU 的两种运行模式。用户态只模拟单一的进程，而非整个系统，允许非本地架构的程序转换为运行主机系统调用的本地指令，优势在于对跨架构的软件开发和软件测试实用性高，例如在 x86 平台上测试编译为 ARM 指令集的程序，局限在于不适合运行涉及多个进程或者需要模拟特定硬件的程序。系统态模拟整个计算机系统，包括 CPU、内存、外围设备等，可以运行包括操作系统在内的软件栈，优势在于能够模拟不同的操作系统和硬件配置，适用于开发、调试和测试软件在多种硬件和系统配置上的行为，局限在于模拟层次较低，可能需要较高的计算性能，并可能存在性能开销问题。Gem5 的全系统模式可以利用 QEMU 来模拟处理器的执行，这种方式融合了两个工具的优势。QEMU 提供了高性能的二进制翻译技术，而 Gem5 提供了更详细和精确的系统模型。

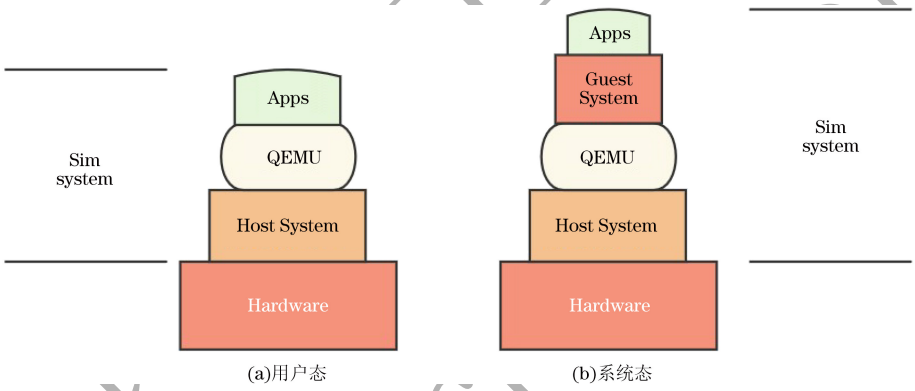

图 2 QEMU 两种运行模式

Fig. 2 Two operation modes of the QEMU

2.4 小结

CPU 作为计算机中最为常见且重要的部件，其模拟器的发展也最为领先，但是仍有其局限性，目前的 CPU 模拟器大多着重于其共性技术，如虚拟化技术、指令的动态翻译技术、多核支持技术等，在协同其他处理器模拟器及其他器件等方面有着明显缺失，无法像真实的 CPU 那样有着强大的协同能力，且在多线程和并发性能方面无法实现真实模拟，模拟器可能无法准确地反映硬件中的并发和资源争用。

3 GPU 模拟器

GPU 的发展紧随 CPU 之后。在 CPU 出现后的几十年里，为了满足计算机图形渲染的需求，科研人员开始了对图形处理技术的探索。早期的 GPU 主要用于图形渲染，直到 2006 年 NVIDIA 发布了首款支持通用计算的 GPU(GeForce 8800 GTX)，通用 GPU 计算才迎来了飞速发展。通用 GPU 的模拟器发展相对缓慢，直到 2009 年 6 月，加拿大英属哥伦比亚大学(UBC)的研究团队发布了世界上第一款支持周期精确的通用 GPU 模拟器 GPGPU-Sim。尽管 NVIDIA 在 HPCA 2015 上介绍了其内部开发的系统级 GPU 模拟器 NVArchSim (NVAS)[22]，其模拟性能实现了数量级的提升，但由于该模拟器并未开源，GPGPU-Sim 仍然是 GPU 研究领域的首选工具。本节将简单介绍 GPGPU-Sim 的实现原理与技术。

3.1 GPGPU-Sim 简介

GPGPU-Sim 是一个开源的 GPU 及通用计算项目，它主要用于模拟和评估 GPU 的性能。它是全球首款周期精确级的 GPGPU 模拟器，是在 SimpleScalar[23] 的基础上实现的，不仅扩展了功能，而且还对性能及结构进行了优化，可以对不同的 GPU 架构进行性能评估、研究和优化。GPGPU-Sim 提供了灵活的模拟环境，可以模拟各种 GPU 的并行计算和内存系统，并且支持多个 GPU 核心、多个内存层级及其互联方式的模拟。它还具备动态优化和调试能力，可以用于研究开发新的 GPU 架构、优化 GPU 程序性能以及进行并行计算系统的研究等。

3.2 GPGPU-Sim 结构

GPGPU-Sim 通过三大模块来实现 GPU 模拟

功能，分别为 GPU 核心模块、互联网络模块、存储模块，如图 3 所示。这三大模块间的互相配合实现了基础的 GPU 功能，下文将介绍这三大模块的具体内容。

1）GPU 核心模块：GPU 核心模块是 GPGPU-Sim 的核心组件，负责解析、调度和执行 GPU 的指令。它包含指令调度器模块和多个流多处理器（SM）。单个 SM 类似于简单 MIPS 五栈顺序的流水线结构，含有寄存器模块、线程调度器模块及多个执行单元。

2）互联网络模块：GPGPU-Sim 中的互联网络模块是模拟 GPU 中多个处理单元之间的通信和数据传输的关键模块。它用于模拟 GPU 中的片上互联和片外互联，在模拟器中用来连接 GPU 核心模块与存储模块，实现两者之间的数据传递。通过互联网络模块，GPGPU-Sim 能够准确地评估模拟出 GPU 中处理单元之间的通信性能和数据传输效率。

3）存储模块：在 GPGPU-Sim 中，存储模块精确地模拟了 GPU 中的存储层级结构和存储访问行为。它使用专门的动态随机存取存储器（DRAM）来进行数据传输的管控，以确保模拟出的 GPU 内存在传输效率上与真实 GPU 内存相当。

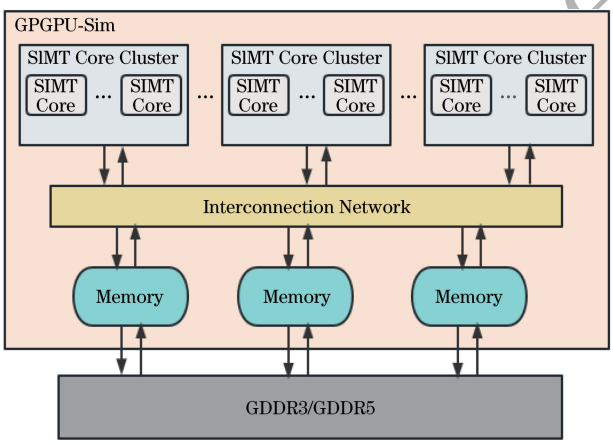

图 3 GPGPU-Sim 结构

Fig. 3 GPGPU-Sim structure

3.3 小结

自 2009 年 GPGPU-Sim 问世以来的十多年里，除了 NVIDIA 内部开发的 NVAS 之外，学术界和工业界尚未发布其他全新的 GPU 模拟器。大多数研究是基于 GPGPU-Sim 的改进或扩展，例如 Gem5-gpu 就是将 Gem5 与 GPGPU-Sim 结合开发的成果。对于 GPGPU-Sim 而言，其局限性主要在于性能模拟及微架构模拟，它无法提供与硬件相同的性能。在微架构层面，GPGPU-Sim 的一些假设可能并不总是精确地反映特定 GPU 的行为。例如，其缓存策略、流水线行为和调度策略可能都是简化的或理想化的。另外，GPGPU-Sim 的更新及维护也是一个较大的挑战，GPU 迭代发展较快，更新模拟器以跟上 GPU 的更新速度是一个较为复杂的可持续性工作。

4 DSP 模拟器

早期的 DSP 芯片基于通用微处理器架构，主要用于实现基本的数字信号处理功能。这类 DSP 芯片成本较低，但由于其应用范围有限，当时对 DSP 模拟器的需求和发展几乎不存在。直到 1982 年，TI 推出了 TMS32010，这是第一款专门用于数字信号处理的单芯片 DSP。TMS32010 的出现成为 DSP 芯片发展的里程碑，也推动了 DSP 模拟器的发展进入新阶段。DSP 模拟器的发展与 DSP 芯片及开发工具的发展密切相关。这类模拟器通常由各公司内部开发，并整合到对应的集成开发环境（IDE）中。例如：作为 DSP 芯片领域的领军企业，TI 发布了 TI CCS，CEVA 公司也推出了 CEVA-Toolbox，这两种工具集成了 DSP 模拟器功能，用于开发和调试 DSP 程序。本节以 TI CCS 与 CEVA-Toolbox 模拟器为例，简单介绍 DSP 模拟器。

4.1 TI CCS 模拟器

TI CCS 是 TI 公司推出的一款 IDE，用于开发和调试基于 TI 处理器的嵌入式系统和应用程序。它是专门为 TI 的 DSP 和微控制器产品系列设计的，提供了一套全面且强大的工具，用于加快嵌入式软件开发的效率和质量。它包括代码编辑器、编译器、调试器、仿真器、性能分析工具等，以及与这些产品配套的软件库和文档。其中仿真器具有模拟器的功能，但只针对 TI 公司芯片的模拟，仿真器可以模拟芯片的指令集、寄存器、内存结构、中断处理等功能。借助 TI CCS 中的片上高速缓存冲突分析器、流水线失速分析器、代码覆盖率分析器和多事件功能分析器，开发人员可以利用仿真器来进行高效的调试和仿真工作，包括设置断点、单步执行代码、观察和修改变量值等。

TI CCS 通过与仿真器的交互，模拟目标 DSP 芯片的行为，使开发人员能够进行实时的调试和分析。需要注意的是，仿真器的模拟精度和功能在很大程度上依赖于目标 DSP 芯片的架构以及 TI CCS 的支持。不同型号的芯片和不同版本的 TI CCS 可能提供不同的仿真和调试功能。因此，在使用仿真

器进行模拟时，应参考针对特定芯片和 TI CCS 版本的文档和指南，了解其具体的模拟能力和限制。由于 TI 的 CCS 开源信息较为有限，因此对于其模拟器的架构和模拟机制并没有太多公开资料可分析。图 4 为 TI C66x 的内核架构图[24]，其可作为模拟器架构的参考。

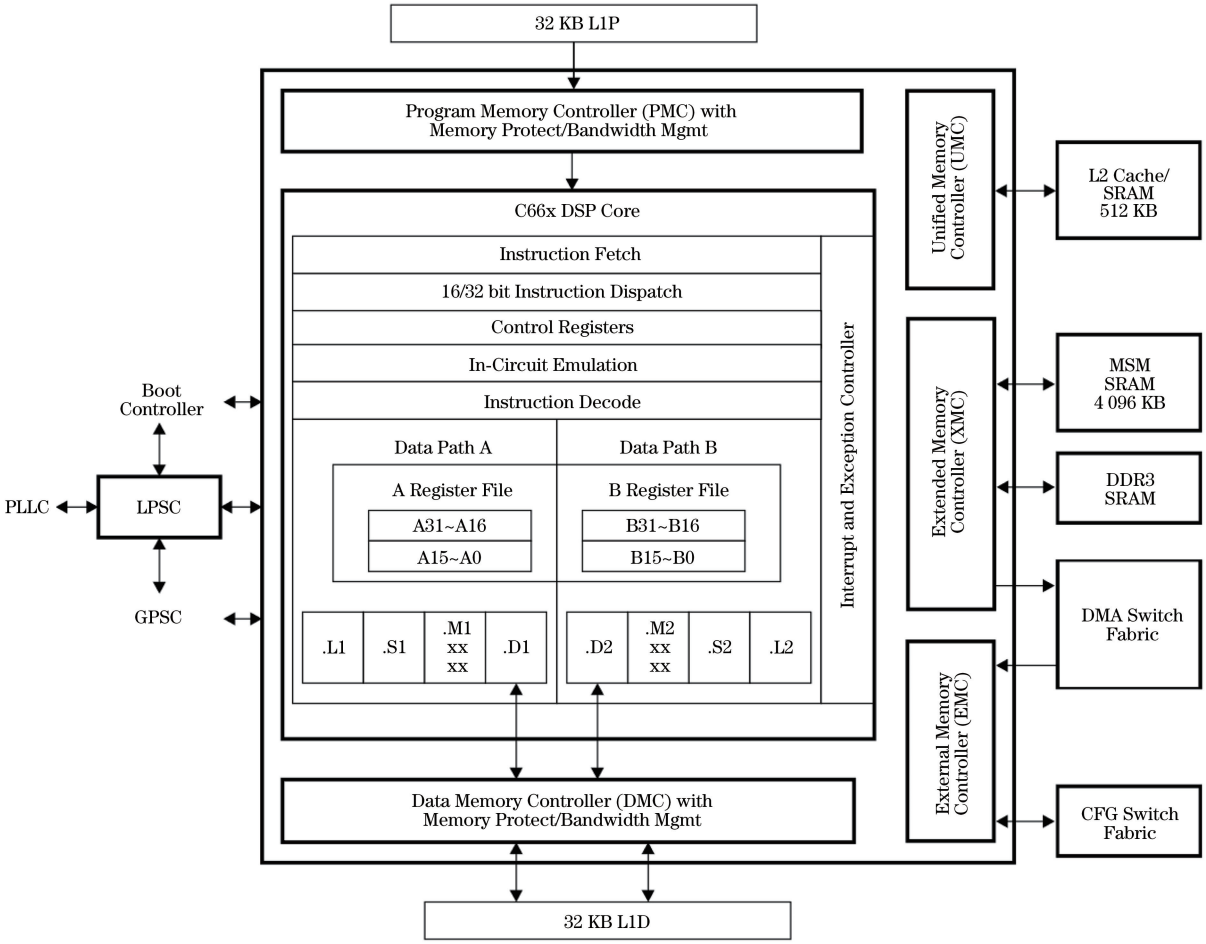

图 4 TI C66x 内核架构

Fig. 4 TI C66x kernel architecture

4.2 CEVA-Toolbox 模拟器

CEVA-Toolbox 是一款由 CEVA 公司开发的专业工具箱，用于设计和开发 DSP 算法和系统。它提供了一系列工具和资源，能够加速 DSP 算法的开发和优化，适用于多种应用领域。在这套工具箱中包含了架构模拟器功能，类似于 TI CCS 中的仿真器，但仅支持模拟 CEVA 旗下的芯片。其架构模拟器的主要特点和功能如下：

1）指令集模拟：架构模拟器可以模拟 CEVA 的 DSP 处理器的指令集，包括指令的执行过程、寄存器操作、内存访问等。它能够详细模拟处理器的各种指令和功能，以及指令执行的时钟周期。

2）微体系结构模拟：架构模拟器能够模拟 CEVA 处理器的微体系结构，包括流水线结构、内存子系统、运算单元等。它可以分析和评估处理器的性能特征，如吞吐量、延迟、并行性等。

3）性能评估：架构模拟器配合 CEVA-Toolbox 中的分析器，通过模拟和分析处理器的行为，可以提供各种性能评估指标。它可以衡量指令执行的效率、内存访问的延迟、功耗等，并帮助优化处理器的架构设计和参数配置。

4）调试和优化：架构模拟器配合 CEVA-Toolbox 调试器和优化工具，可以帮助开发人员分析和解决处理器设计中的问题。通过模拟执行和追踪指令的执行过程，可以检测潜在的性能瓶颈、数据冲突和错误行为，并进行相应的调整和优化。

由于 DSP 的发展历程与 CPU 不同，缺乏长时间的技术交流和开源社区的支持，不同公司生产的 DSP 芯片采用不同的架构与指令集。正如前文所述，不同公司的开发工具之间并不兼容，且目前也没有开源的通用模拟器供研究社区使用。这种现状导致了 DSP 模拟器领域的局限性。随着 DSP 技术的发展，未来对于开源的通用 DSP 模拟器的需求将会越来越强烈。

5 神经网络处理器模拟器

随着深度学习技术的快速发展，研究人员开始意识到传统的 CPU 和 GPU 在执行神经网络计算时存在性能瓶颈。为了解决这个问题，研究人员开始提出一些专用的神经网络加速器。随着越来越多的公司和学术机构开始研发神经网络处理器，这一领域取得了重大的技术突破。例如：NVIDIA 推出了一系列针对深度学习的 GPU，如 Tesla 和 Tegra 系列；Google 也开发了张量处理单元（TPU），专门用于加速神经网络计算。随着这些新型处理器的出现和发展，与之对应的模拟器技术也得到了显著进步，例如：ARM 研究院与波士顿大学和佐治亚理工学院合作开发了 SCALE-Sim；佐治亚理工学院与英伟达合作开发了 MAESTRO。本节以这两种模拟器为例，介绍神经网络处理器模拟器。

5.1 SCALE-Sim 模拟器

SCALE-Sim 是一个用于模拟和评估深度学习加速器的开源工具，旨在帮助研究人员和工程师进行深度学习模型和硬件体系结构的协同设计，并提供性能和功耗评估，其架构简图如图 5 所示。SCALE-Sim 的特点和功能包括：

1）可扩展性：SCALE-Sim 能模拟各种规模的加速器体系结构，从微型加速器到大型数据中心加速器。

2）灵活的配置：SCALE-Sim 允许用户根据自己的需求灵活配置加速器的结构和参数，包括处理单元数量、存储容量、总线带宽等。

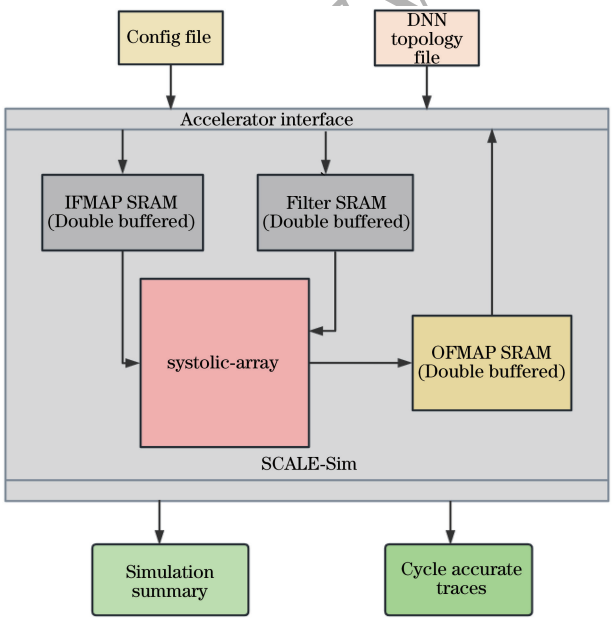

图 5 SCALE-Sim 结构简图

Fig. 5 Diagram of SCALE-Sim structure

3）多样化的工作负载：SCALE-Sim 支持多种深度学习工作负载的仿真，包括卷积神经网络、循环神经网络等。

4）性能和功耗分析：SCALE-Sim 提供了各种性能指标，如推理时间、帧率、吞吐量等，它们用于评估加速器的性能以及估算功耗和能耗。

SCALE-Sim 是一种基于收缩阵列架构的模拟器，是第一款开源的可配置的支持精确周期仿真的 CNN 加速器模拟器。它的出现为神经网络处理器模拟器的发展开创了新的起点。

5.2 MAESTRO 模拟器

MAESTRO 是一款专为深度学习加速器设计的开源工具，旨在促进对不同数据流方案的理解和分析。它将数据流建模为循环的多维数组，这些数组代表了加速器内部的各种操作，包括读取、计算和写入。MAESTRO 的目标是为研究人员和开发人员提供一个评估微架构设计效果的框架。MAESTRO 的主要功能与特点如下：

1）领域特定语言（DSL）：MAESTRO 的 DSL 被用来描述深度学习操作和数据流方案。它为向量化操作的所有参数提供了定义，模型化了缓存，并考虑了数据复用和内存层次结构。通过使用 DSL，MAESTRO 提供了对深度学习操作和数据流方案的简单直接的描述和分析。

2）数据流成本效益分析框架：MAESTRO 接收由 DSL 编写的数据流，将神经网络维度和硬件资源信息作为输入。该框架能够生成输入数据流的活动次数统计、硬件资源需求和性能极限吞吐量报告。此外，MAESTRO 还考虑了各类数据复用模式，以确定精确的缓冲区访问和大小需求，并支持通信延迟分析。

3）加速器信息参考：MAESTRO 提供部分加速器（如 Eyeriss、NVDLA 等）的数据流之间的权衡分析，并支持用户根据可用的硬件资源和目标神经网络形状进行配置。随着开源工作的发展，MAESTRO 将会集成更多加速器的相关信息，从而减少冗余的配置工作。

通过提供开源的配置、广泛的应用、抽象的数据流建模、过程分析和评估工具以及性能和能效标准，MAESTRO 能够帮助研究人员和开发人员更高效地设计和优化加速器以实现更好的性能和能效。通过 DSL，MAESTRO 进一步简化了对深度学习操作和数据流方案的描述和分析，使得用户可以进行定制化的设计，以满足特定的性能、能效和存储需求。

5.3 小结

神经网络处理器作为近几年的热门研究方向，吸引了众多研究机构和企业的关注，推动了多种神经网络处理器模拟器的快速发展。这些模拟器的模拟技术大致可分为以数据为中心的模拟、以计算为中心的模拟、以关系为中心的模拟[25]，主要提供性能分析及功耗分析功能。然而，它们在加速性能模拟及微体系结构模拟方面仍存在较大的局限性。

6 体系结构模拟器挑战及展望

表 1 对上述体系结构模拟器的部分特点进行对比。结合是否具有可扩展性与是否开源两项可知，一款模拟器的扩展性及发展程度大部分取决于开源与否，一款优秀的模拟器需要计算机领域的众多科研人员共同努力，任何一个领域的发展也离不开众多开源软件的支持。

表 1 体系结构模拟器对比

Table 1 Architectural simulator comparison

模拟器	是否可扩展	模拟层次	是否开源	架构支持
QEMU	可扩展	系统/指令级	是	CPU
Gem5	可扩展	系统/微体系结构/指令级	是	CPU
GPGPU-Sim	可扩展	微体系结构级	是	GPU
TI CCS	不可扩展	微体系结构级	否	DSP(TI 系列)
CEVA-Toolbox	不可扩展	微体系结构级	否	DSP(CEVA 系列)
MAESTRO	可扩展	微体系结构级	是	AI 芯片
SCALE-Sim	可扩展	微体系结构级	是	AI 芯片

表 1 中的模拟层次反映了不同模拟器的侧重点。系统级模拟的优势在于可以模拟完整的系统，适用于操作系统和系统软件的开发与测试，局限性在于不能提供指令的精确模拟且在精密性能评估上不够准确。指令级模拟精确度高，适用于硬件开发早期阶段，但运行速度慢、运行开销较大。微体系结构级模拟的优势在于更关注硬件的内部结构，适用于硬件微体系结构设计的验证和优化，可进行详细的性能分析，局限性在于模拟器设计复杂，需要长时间的研发，并且其运行速率更慢、运行开销更大，难以模拟大规模的系统。通过对模拟层次的分析可知，目前大部分的模拟器的模拟层次是微体系结构级，其底层实现技术是根据硬件的结构来实现的，从而模拟出硬件的微体系结构，例如，数据的传输与存储模式，处理单元的运行算法。从架构支持来看，目前能够支持多种处理器架构的模拟器仍然较少。Gem5 在 GPU 模拟器方面已经迈出了重要的一步，但对于能够支持多种异构架构的模拟器尚未有明显的发展趋势。结合上述分析及未来处理器的演变方向，本文从模拟器的角度出发，提出对体系结构模拟器的几点挑战和展望。

目前，体系结构模拟器面临的主要挑战具体如下：

1)开源是目前计算机领域发展的重要趋势。全球现阶段对于开源理念的推崇达到了前所未有的高度，对于模拟器软件同样有着十分重要的影响。随着计算机技术的持续发展，未来会出现各种新型处理器，同样也会需要对应新架构的新型模拟器。然而，模拟器的新架构的后端开发支持并不是一件高效便捷的工作，尤其对于架构差异较大的专用处理器。因此，实现模拟器后端的开发支持是未来的一个发展方向，这需要一个良好的开源社区以及所有科研人员一起努力，共同推进模拟器的发展。

2)开发一款能够支持多种异构架构的模拟器已经迫在眉睫。如今，随着各种新型应用的不断涌现，对于算力的需求日益增长，传统的并行、多核、异构等架构已经无法满足需求。为了提高算力，Intel 公司已经提出了超异构的概念[26]，未来处理器体系结构将朝着多元化、超异构融合的方向发展。对于模拟器而言，如何适配异构乃至超异构的架构模拟，使其更贴近真实的芯片环境与性能，将成为一项重要的挑战，同时解决这一挑战也将推动模拟器与处理器发展直接形成良好的反馈循环，促进技术的共同进步。

3)相比于真实硬件的性能提升，模拟器与其之间仍存在显著的差距。目前，模拟器在微体系结构及数据传输等技术的实现上，大多仿照真实硬件的结构及数据传输模式。然而，对于需要模拟性能的模拟器而言，仅仿照真实硬件的实现方式难以实现有效的性能提升。因此，模拟器需要在技术上实现重大突破，研发出新型高效的实现方法，以弥补其在性能提升上的不足。

笔者对于体系结构模拟器的未来展望具体如下：

1)若要避免处理器更新换代带来的模拟器重复

性开发工作,则亟需研发一款通用模拟器。基于上述分析,不管哪种类型的模拟器,其基础的架构都是通过仿照实际的硬件架构来设计的。为了实现模拟器的开发支持,可以借鉴编译器的一般架构,将处理器中的各种底层硬件操作抽象化,构建形成统一的中间描述,上层利用中间描述的搭配组合实现不同的处理器架构。这种方式不仅可以避免重复开发,减少在不同处理器架构模拟上的冗余工作,而且还能推动形成一个通用的体系结构模拟器框架。

2)模拟器想要实现较高的性能提升,不仅需要技术的革新,而且还应充分利用宿主机的硬件优势。然而,在本文讨论的模拟器中,很少有考虑到发挥宿主机硬件特性的设计。对于模拟器性能的提升,仅依靠架构优化已经难以取得显著进展。如果将性能提升的思路转向利用宿主机的硬件优势,可能会取得更好的效果。例如,在 x86 架构的宿主机上,如果能够将模拟的向量计算指令转化为 x86 的向量扩展指令[如 SSE(Streaming SIMD Extensions)、AVX(Advanced Vector Extensions)等],可以显著提升模拟器的性能。

7 结束语

在提倡节约资源的当今社会,模拟器在科研领域中的地位越来越高,已成为许多领域发展不可或缺的工具。本文分别介绍了 CPU、GPU、DSP、神经网络处理器这 4 种主流处理器及其对应的主流模拟器。根据不同模拟器的主要特点与发展状况,重点讨论了模拟器面临的主要挑战与未来的研究方向。

模拟器技术和体系结构发展息息相关。随着体系结构的进步,尤其是专用处理器架构的不断创新,模拟器需要为其提供基础平台支持,并为体系结构设计提供反馈。未来模拟器技术的研究可以从以下几个方面展开:后端敏捷开发,宿主机硬件优势利用,通用模拟架构设计以及超异构架构模拟。下一步本团队将基于 QEMU 开展对 DSP 架构支持的研究。

参考文献

[1] 李祥兵. 基于 GNU Binutils 的嵌入式系统交叉汇编器和交叉连接器的移植[D]. 杭州:浙江大学, 2004.
LI X B. Transplantation of cross assembler and cross connector for embedded system based on GNU Binutils[D]. Hangzhou: Zhejiang University, 2004. (in Chinese)

[2] GPGPU-Sim_Distribution[EB/OL]. [2023-10-05]. https://github.com/gpgpu-sim.

[3] RILEY G F, HENDERSON T R. The NS-3 network simulator[M]. Berlin, Germany: Springer, 2010.

[4] 张乾龙, 侯锐, 杨思博, 等. 体系结构模拟器在处理器设计过程中的作用[J]. 计算机研究与发展, 2019, 56(12): 2702-2719.
ZHANG Q L, HOU R, YANG S B, et al. The role of architecture simulators in the process of CPU design[J]. Journal of Computer Research and Development, 2019, 56(12): 2702-2719. (in Chinese)

[5] 许建卫, 陈明宇, 杨伟, 等. 计算机体系结构模拟器技术和发展[J]. 系统仿真学报, 2009, 21(20): 6325-6331.
XU J W, CHEN M Y, YANG W, et al. Technology and trends of computer architecture simulators[J]. Journal of System Simulation, 2009, 21(20): 6325-6331. (in Chinese)

[6] AKRAM A, SAWALHA L. A survey of computer architecture simulation techniques and tools[J]. IEEE Access, 2019, 7: 78120-78145.

[7] 鄢贵海, 卢文岩, 李晓维, 等. 专用处理器比较分析[J]. 中国科学: 信息科学, 2022, 52(2): 358-375.
YANG H, LU W Y, LI X W, et al. Comparative study of the domain-specific processors[J]. Scientia Sinica (Informationis), 2022, 52(2): 358-375. (in Chinese)

[8] BELLARD F. QEMU, a fast and portable dynamic translator[C]// Proceedings of USENIX Annual Technical Conference. New York, USA: ACM Press, 2005: 41-46.

[9] BINKERT N, BECKMANN B, BLACK G, et al. The Gem5 simulator[J]. ACM SIGARCH Computer Architecture News, 2011, 39(2): 1-7.

[10] 刘雨辰, 王佳, 陈云霁, 等. 计算机系统模拟器研究综述[J]. 计算机研究与发展, 2015, 52(1): 3-15.
LIU Y C, WANG J, CHEN Y J, et al. Survey on computer system simulator[J]. Journal of Computer Research and Development, 2015, 52(1): 3-15. (in Chinese)

[11] LAWTON K P. Bochs: a portable PC emulator for Unix/X[J]. Linux Journal, 1996(29): 7-8.

[12] POWER J, HESTNESS J, ORR M S, et al. Gem5-gpu: a heterogeneous CPU-GPU simulator[J]. IEEE Computer Architecture Letters, 2015, 14(1): 34-36.

[13] PATIL H, PEREIRA C, STALLCUP M, et al. PinPlay: a framework for deterministic replay and reproducible analysis of parallel programs[C]// Proceedings of the 8th Annual IEEE/ACM International Symposium on Code Generation and Optimization. New York, USA: ACM Press, 2010: 2-11.

[14] TAN Z X, WATERMAN A, AVIZIENIS R, et al. RAMP gold: an FPGA-based architecture simulator for multiprocessors[C]//Proceedings of the Design Automation Conference. Washington D. C., USA: IEEE Press, 2010: 463-468.

[15] COLLANGE C, DAUMAS M, DEFOUR D, et al. Barra: a parallel functional simulator for GPGPU[C]//Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems. Washington D. C., USA: IEEE Press, 2010: 351-360.

[16] CCSTUDIO[EB/OL]. [2023-10-05]. https://www.ti.com.cn/tool/cn/CCSTUDIO? keyMatch=CCS#tech-docs.

[17] CEVA-Toolbox[EB/OL]. [2023-10-05]. https://www.ceva-dsp.com/wp-content/uploads/2021/03/02_19_21_ToolBox_Product_Note_EN-V4_SC.pdf.

[18] KWON H, KRISHNA T. MAESTRO: an open-source infrastructure for the cost-benefit analysis of dataflows within deep learning accelerators[EB/OL]. [2023-10-05]. https://github.com/maestro-project/maestro.

[19] SAMAJDAR A, ZHU Y H, WHATMOUGH P, et al. SCALE-Sim: systolic CNN accelerator simulator[EB/OL]. [2023-10-05]. https://arxiv.org/abs/1811.02883v2.

[20] BINKERT N L, DRESLINSKI R G, HSU L R, et al. The M5 simulator: modeling networked systems[J]. IEEE Micro, 2006, 26(4): 52-60.

[21] MARTIN M M K, SORIN D J, BECKMANN B M, et al. Multifacet's General Execution-driven Multiprocessor Simulator (GEMS) toolset[J]. ACM SIGARCH Computer Architecture News, 2005, 33(4): 92-99.

[22] VILLA O, LUSTIG D, YAN Z, et al. Need for speed: experiences building a trustworthy system-level GPU simulator[C]// Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA). Washington D.C., USA: IEEE Press, 2021: 868-880.

[23] AUSTIN T, LARSON E, ERNST D. SimpleScalar: an infrastructure for computer system modeling[J]. Computer, 2002, 35(2): 59-67.

[24] Texas Instruments. Multicore fixed and floating-point digital signal processor[EB/OL]. [2023-10-05]. https://www.ti.com/lit/ds/symlink/tms320c6678.pdf.

[25] 李宇航. 面向"迈动"加速器核的模拟器研究与设计[D]. 长沙: 国防科技大学, 2022.
LI Y H. Research and design of simulator for "Maidong" accelerator core[D]. Changsha: National University of Defense Technology, 2022. (in Chinese)

[26] 宋继强. AI 计算迈入超异构时代[EB/OL]. [2023-10-05]. https://36kr.com/p/1723494694913.
SONG J Q. AI computing has entered the era of hyper-heterogeneity[EB/OL]. [2023-10-05]. https://36kr.com/p/1723494694913. (in Chinese)

编辑 陆燕菲