

DOI:10.3969/j.issn.1003-5060.2025.03.007

一种软硬件协同的多核系统模拟器的设计与实现

王 鉴, 倪 伟

(合肥工业大学 微电子学院, 安徽 合肥 230601)

摘 要:基于软件实现的多核系统模拟器执行计算密集/数据密集任务的时效性极差,且存在模拟精度和性能评估准确性差的不足,限制其在多核系统结构优化探索中的应用。文章提出一种周期精确的软硬件协同多核系统模拟器(cycle accurate hardware-software co-simulator, CAHSCS),通过在传统模拟器架构中引入硬件计算和存储模块,CAHSCS 能有效改善全系统的模拟速度、精度,提高性能评估的准确性。复杂真实任务加载实验结果表明,CAHSCS 将大规模复杂数据的运算效率提高了 10 倍,显著加快了系统设计收敛速度。

关键词:软硬件协同模拟器;多核系统模拟器;大规模数据运算;硬件计算;模拟器加速

中图分类号:TN431.2 **文献标志码:**A **文章编号:**1003-5060(2025)03-0335-08

Design and implementation of a multi-core system simulator based on hardware-software synergy

WANG Jian, NI Wei

(School of Microelectronics, Hefei University of Technology, Hefei 230601, China)

Abstract: Software-based multi-core system simulators have extremely poor timeliness in performing computation-intensive/data-intensive tasks, and have poor simulation accuracy and performance evaluation accuracy, which limits their application in the exploration of multi-core system structure optimization. This paper proposes a hardware-software cooperative multi-core system simulator called cycle accurate hardware-software co-simulator(CAHSCS). By introducing hardware computing and storage modules into the traditional simulator architecture, CAHSCS effectively improves the simulation speed and accuracy of the entire system and the accuracy of performance evaluation. The complex real task loading experiment shows that CAHSCS improves the computing efficiency of large-scale complex data by ten times, and significantly speeds up the convergence speed of system design.

Key words: hardware-software co-simulator; multi-core system simulator; large-scale data computing; hardware computing; simulator acceleration

0 引 言

多核系统的诞生为硬件电路的设计提供了更广阔的空间,而更复杂的芯片结构也对系统验证提出更加苛刻的要求。系统级建模是多核系统设计流程中的重要一环,目的在于快速获得目标系统的相关数据,为后续的设计工作提供强有力的

支持。当前多核系统模拟器多为软件实现方式,已公开报道的相关工作主要如下:

1) 在功能模块研究工作方面。文献[1]研究了无限芯片网络的模型在基于经典多核系统模拟器 Noxim 上完成架构实现和评估,在功耗方面获得一定程度的优化;文献[2]指出超大规模硬件系统在设计之初,会采用混合精度建模与硬件原型

收稿日期:2023-03-29;修回日期:2023-04-25

基金项目:安徽省高校协同创新资助项目(GXXT-2019-030)

作者简介:王 鉴(1996—),男,湖北武汉人,合肥工业大学硕士生;

倪 伟(1977—),男,安徽合肥人,博士,合肥工业大学副教授,硕士生导师,通信作者,E-mail:ni.wei@hfut.edu.cn.

协同验证方案;文献[3]在 BookSim 中添加若干模拟工具,通过应用程序编程接口将 BookSim 集成到 gem5 中,实现模拟器的功能扩展,并与 Garnet 模型进行比较,同时建立旁路模拟工具集,优化片上网络通信;文献[4]介绍了使用人工案例测试模拟器的可扩展性瓶颈,通过可扩展性测量模型使用阿姆达尔定律进行强扩展性测量。

2) 在新算法探索方面。文献[5]围绕量子退火算法设计的模拟器对不同 Ising 模型进行了研究分析,并对相关算法实现现场可编程逻辑门阵列(field programmable gate array, FPGA)加速;文献[6]研究了可重构计算机硬件内部电路结构与外部接口的耦合以及将高级算法映射到可重构底层硬件的方式,阐述了 FPGA 构成异构系统的基本方式和原理,重点讨论软件方式描述硬件的过程与实现,为硬件设计提出指导;文献[7]在设计阶段为多核控制软件设计了仿真环境,通过在控制程序中加入二级核心控制结构,从而使模拟系统控制行为更接近实际功能模拟的细节;文献[8]提出一个框架对应用程序行为进行检测分析,并提供数据处理的缓存信息,设计缓存锁定选择算法将数据锁定在缓存中,用于优化程序的可预测性。

3) 在模拟器实现方面。文献[9]通过汇编程序集的翻译和测量指令执行时间,使用多线程函数模拟出嵌入式操作系统的多核相关的应用程序编程接口(application programming interface, API),优化了 RH850 和 Autosar 操作系统;文献[10]设计了一个多核并行的脉冲神经网络模拟器对神经元进行编码和映射,通过自定义路由表解决了多核间的网络通信,以时间驱动为策略实现了核与核间的动态同步,并在模拟器上进行了脉冲神经网络的并行计算;文献[11]提出一种面向可重构多核系统的仿真加速方案,将大规模矩阵运算和大点数快速傅里叶变换(fast Fourier transform, FFT)等任务分解为多个子任务,并利用子任务内部以及不同子任务之间在指令和数据操作上的强耦合特性,实现了一定的运算加速效果;文献[12]实现了一种系统级异构多核混合精度可编程模拟器,以支持超大规模和可配置结构模型。

目前上述研究工作重点在于丰富模拟器的功能、拓展被模拟的结构以及在软件层面优化模拟器的工作效率、探索新的内部通信技术,关于加速模拟器运行数据/计算密集型任务速度的报道

很少,而执行此类任务得到的系统工作数据是开展系统优化工作的基础。

针对上述问题,本文将硬件加速技术引入模拟器,提出一种软硬件协同模拟器架构,并设计一种周期精确的软硬件协同多核系统模拟器(cycle accurate hardware-software co-simulator, CAHSCS),由运行在 PC 上的软件模拟器和运行在 FPGA 上的硬件加速器组成。对加载真实任务的实测结果进行对比,得出本文模拟器执行数据/计算密集型任务速度是纯软件模拟器的 10 倍。

1 软硬件协同模拟器设计

1.1 CAHSCS 架构概述

典型的基于片上网络的多核系统结构如图 1 所示,由挂载在片上网络上的主控制器、功能单元(存储单元、运算单元等)组成。

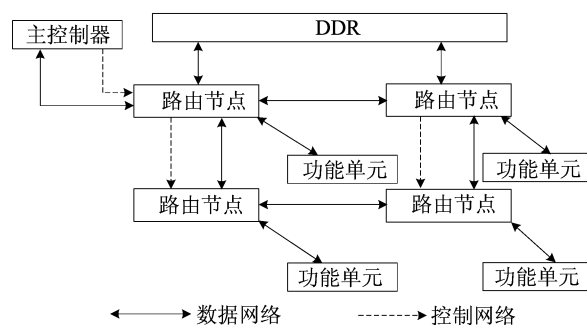


图 1 基于片上网络的多核系统结构示意图

图 1 所示的各部件映射至软硬件协同模拟器框架下,即可得到 CAHSCS 的结构,如图 2 所示。CAHSCS 包括部署在 PC 上的软件模拟器(执行如数据通讯、任务分配等非数值计算任务)、部署在 FPGA 芯片上执行计算加速任务的硬件加速器和连接两者的通用接口 3 个部分。

软件模拟器用 C++ 实现,主功能是模拟片上网络行为(包括可配置的拓扑和节点数量),达到模拟不同多核系统的目标。软件模拟器在本文中用于实现维度顺序路由、 8×8 的 Mesh 拓扑结构和 PCC 网络协议功能,同时管理全模拟器运行、软硬件之间任务同步以及模拟器状态信息额收集统计。

硬件加速器包括与 AXI 总线模块相连的运算簇模块和本地控制模块。本地控制模块根据软件模拟器下发的信息管理硬件加速器运行;运算簇模块集成有若干运算 IP(intellectual property)硬核和对应存储器,这些 IP 与目标多核系统相

同,用于加速大数据复杂任务执行。

通用接口是软件模拟器与硬件加速器之间的

数据和控制通道,用于完成跨平台的控制下发和状态上传以及运算数据与各种被观测信息交互。

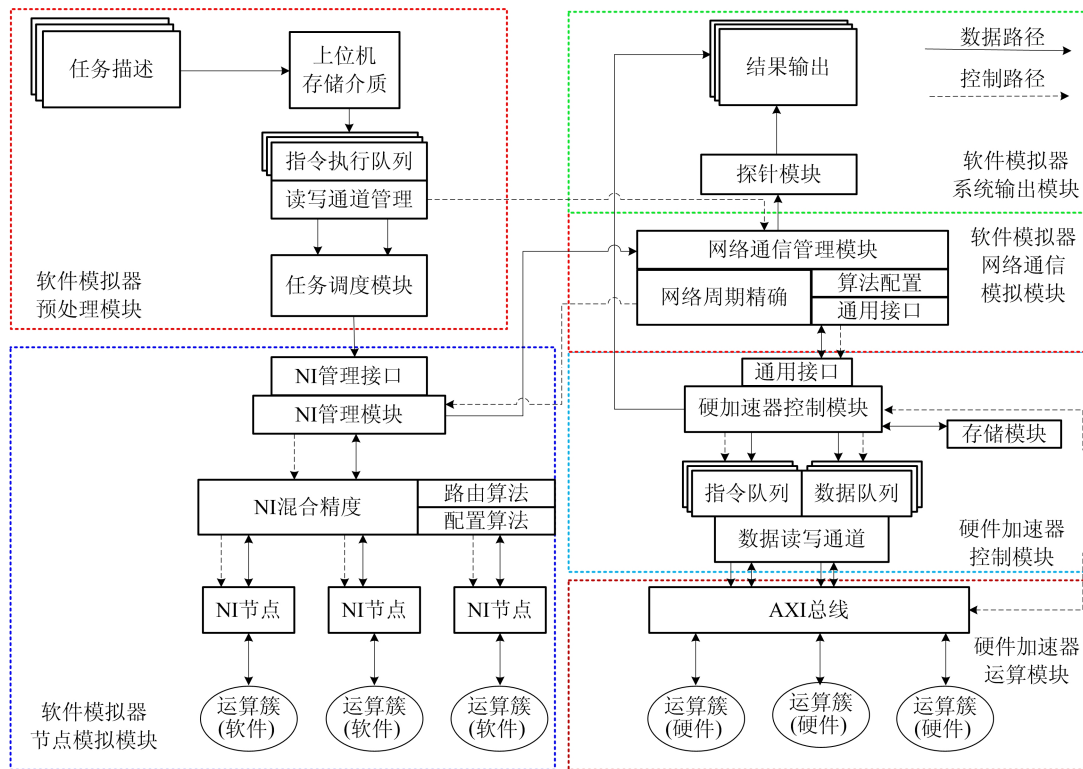


图 2 CAHSCS 结构设计

1.2 软件模拟器设计

CAHSCS 的软件模拟器由 1 个周期精确的片上网络行为模拟器和系统辅助功能模块组成,包括软件模拟器预处理模块、节点模拟模块、系统输出模块等,各模块功能如图 3 所示。

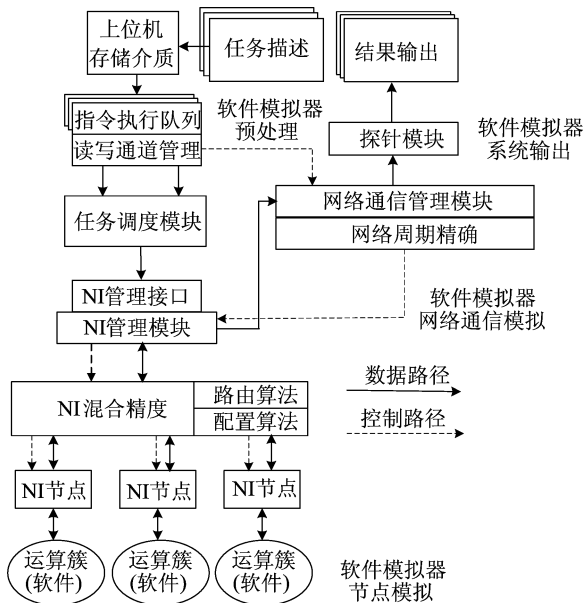


图 3 CAHSCS 软件模拟器结构设计

1) 软件模拟器预处理模块。将输入的多核系统行为模型翻译为工作配置文件,内含软件模拟器工作信息、硬加速器工作信息以及待处理的原始数据信息等。

2) 软件模拟器节点模拟模块和网络通信模拟模块。遵循给定网络协议、路由功能和拓扑结构模拟核间数据通信行为,在网络模块内部的关键部位设置时钟探针,对运行时间进行记录、统计。

3) 软件模拟器系统输出模块。它是运行结果输出端口,模拟器运行数据包括网络行为记录、运算核工作状态记录、任务运算结果和中间数据等。

CAHSCS 的编码模块将任务信息映射至编码数据包内;软件模拟器和硬件加速器中的解码模块对数据包进行解码,从而获得数据包中的任务描述等信息。本文设计了用于系统内部通信的 IP 编码,见表 1 所列。

表 1 为 CAHSCS 中硬件加速器的 FFT 计算单元模块编码规范,可重构计算单元(reconfigurable calculate unit,RCU)模块和通用浮点处理器(general floating-point processor,GFP)模块均

有各自的编码,不同的 IP 使用编码在细节上存在差异,但整体类似。这里以 FFT 硬件 IP 编码为例介绍如下:① 操作码表示使用硬件类型;② 操作组表示当前数据在整体运算中的执行顺序;③ 操作方位表示对映射写入寄存器的编号;④ 节点坐标表示当前 NI(network interface)节点的坐标;⑤ 操作数表示任务的原始数据。

表 1 统一信息编码的 IP 编码

操作码	操作组	操作方位	节点坐标	操作数
31:30	29:28	27:26	25:23	22:0

每个部件都遵从相同的编解码规则,处理任务时对编码数据进行解码,根据信息执行相关数据操作。

1.3 硬件加速器设计

部署在 FPGA 芯片上的硬件加速器是 CAHSCS 的核心,其结构如图 4 所示。硬件加速器包含硬件加速器控制模块、运算模块及模块间实现通信的 AXI(advanced extensible interface)总线 3 个部分。

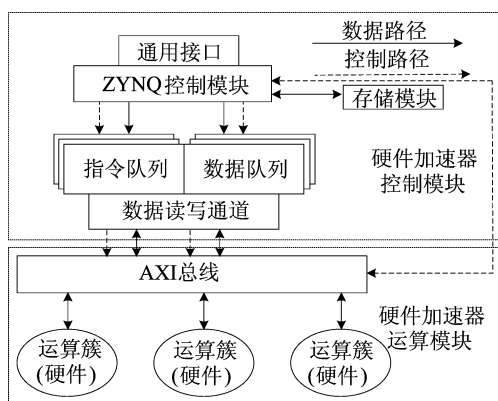


图 4 CAHSCS 硬件加速器结构设计

CAHSCS 的载体为 Xilinx 公司的 ZYNQ 芯片,以 ARM(advanced RISC machine)为控制模块核心,通过软件控制网口单元、存储单元、软硬件交互单元确保硬件加速器协调工作;AXI 总线模块实现各个模块之间的数据传输、数据地址传输、控制信号传输;运算簇模块实现数值运算和部分运算数据存储的功能。

1.3.1 控制模块

硬件加速器控制模块的功能由 ARM 编程实现,包括检索软件模拟器下发的任务文件、进行新任务查询、解析任务与产生编码、管理内部各功能模块完成数据处理任务。

硬件加速器的任务文件包含任务执行顺序,由此可分时复用有限硬件资源完成大规模运算的处理执行。

硬件加速器接收到任务描述编码后立即通过解码模块逐条解析指令并执行。为了保证跨平台数据处理的及时性,硬件加速器维持运行状态,对原始数据文件名进行检测,当未检测到原始数据字段的结果数据文件时,硬件加速器判断存在待处理数据,开始执行相关数据操作。

每次硬件加速器检测到数据处理任务结束后,将结果文件发送至软件模拟器,并存储于指定位置。

1.3.2 运算簇模块

本文运算簇模块集成 FFT、GFP、RCU^[13]等计算单元各 1 个,工作在存储计算模式下,每次运算执行过程为节点从存储器读原始数据、执行运算、节点向存储器写回结果 3 个步骤,同时将每次任务的执行周期回写至硬件加速器控制模块内部任务周期记录部件。

1.4 通用接口设计

CAHSCS 的通用接口结构如图 5 所示。通用接口采用模块化设计,具有良好的平台兼容性,可适配多种类型的软件模拟器平台与硬件加速器载体之间的异构通信。该接口体系由分布式状态机架构和标准化通信协议构成,其中状态机模块分别部署于软件层和硬件层,通过统一的接口规范实现跨平台的高效数据交互。

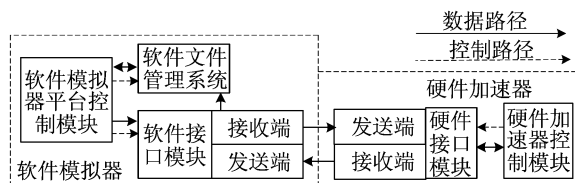


图 5 通用接口结构设计

CAHSCS 规定软件模拟器与硬件加速器的通信数据以文件形式存储。本文设计中软件模拟器与硬件加速器物理通信接口采用基于 UDP 协议的网口实现。软、硬件部分包含解码器,对接收数据包进行解码。将能够解码的数据进一步处理,无法解码的数据舍弃,保证数据传输的安全性。

1.5 CAHSCS 工作流程

CAHSCS 整体工作流程包括软件模拟器执行、软硬件协同工作和硬件加速器工作。多核通信模拟通过软件模拟器执行,包括任务配置、网络

时序模拟、NI 节点运行模拟等。

1.5.1 CAHSCS 软硬件协同工作流程

CAHSCS 软硬件协同运行部分的工程流程如图 6 所示。

NI 节点运行优先判断周期计数器是否为 0,若周期计数器为 0,则跳出当前状态机循环;若周期计数器为非 0,则执行 PCC(policy and charaging control)网络状态的同时,执行周期计数器减 1。上述过程是图 2 中软件模拟器 NI 管理模块、软件模拟器网络通信管理模块和硬件加速器控制模块的协同工作过程。

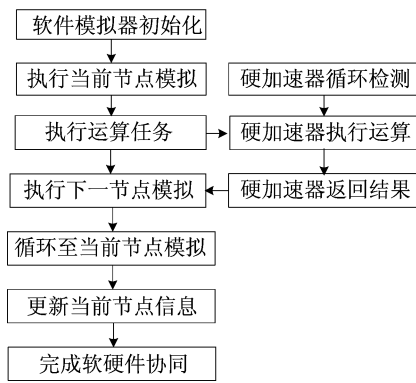


图 6 CAHSCS 软硬件工作流程

PCC 网络完成单次全节点遍历后,由软件模拟器的主控制器依次读取硬件更新数值,按照协议规则更新各个节点的周期计数器;模拟器完成节点内数据处理所消耗的周期数值,将硬件加速器的运算结果进行读取存储于上位机,上述过程映射至图 3 中软件模拟器网络通信管理模块,并将数据传送至 NI 管理模块,通过硬件加速器得出真实计算数据,取代由软件模拟器模拟的部分。

当软件部分的任务计数器数值为 0 时,若所有节点内没有未完成处理的数据且硬件部分状态为 IDLE 时,则输出相应的结果数据和中间过程数据发送至上位机。任务发送片段如图 7 所示。

```

00000A00 <GFP_REG_INPUT_A6>:
A10:  AF793588    reg_3    793588
A14:  AF1FE47D    reg_3    1FE47D
A18:  AF1BCCC9    reg_3    1BCCC9
A1C:  AF160AEF    reg_3    160AEF
A20:  AF21793A    reg_3    21793A
A24:  AF76EF71    reg_3    76EF71

```

图 7 任务发送片段

处理运算任务时,CAHSCS 软件模拟器部分编码规则将任务描述编译为图 7 的形式,并发送至硬件加速器的存储部分;硬件加速器的解码模块将任务进行解码并逐条执行完成任务。图 7 中数据格式按照表 1 规则进行编码实现,依次表示任务序列数、任务信息-1、硬件加速器寄存器序号和任务信息-2,例如 A10 指令为向 GFP 的 2'b10 也就是 reg_3 寄存器中写入数值 0X793588。

1.5.2 硬件加速器工作流程

硬件加速器的工作流程如图 8 所示。原始数据依次通过硬件加速器存储介质、指令队列/数据队列、读写通道模块、任务调度模块、运算执行模块、数据存储/管理模块、数据发送模块。

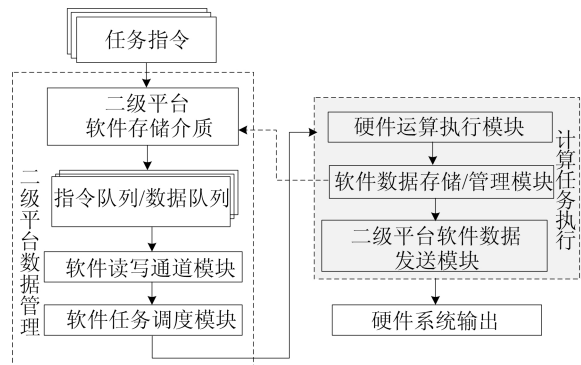


图 8 硬件加速器工作流程

硬件加速器控制模块的程序主功能为存储器检测、解码下发、存储器写回。控制模块循环运行状态机,查询是否存在未执行任务;检测到未执行任务后,启动硬件加速器的运算执行程序;运行程序将未执行的任务进行读取并解码,通过写入目的寄存器完成任务的下发,并保持对硬件加速器中运算簇寄存器的监听,将结果数据读取并存储至存储模块,同时对结果文件命名。

硬件加速器中运算簇存在大量可读寄存器,用于反馈运算簇的状态信息;硬件加速器控制模块的串口模块对模块状态信息进行配置,并打印配置结果用于 Debug 和调试。

2 实验测试

CAHSCS 在保持传统模拟器使用灵活的同时,有效提高了执行计算密集型和数据密集型任务的效率。CAHSCS 验证工作如下:首先验证 CAHSCS 软件模拟器对不同类型网络行为的模拟能力;其次验证 CAHSCS 对 FFT、向量乘法及其组合的加速能力;最后选择 RD SAR 算法的距

离压缩算法验证 CAHSCS 对大数据量、复杂计算任务的加速能力。

2.1 测试环境

本文在 Visual Studio 2019 上使用 C++ 完成 CAHSCS 软件模拟器的建模,使用 Verilog HDL 完成硬件加速器的设计。软件模拟器的的工作平台是 PC, CPU 为 Intel i7-8700 @3.2 GHz, 内存为 32 GiB DDR4@1 600 MHz;硬件加速器在 Xilinx ZYNQ 系列的 Z-7010 板上运行,芯片为 XC7Z010CLG400-1,内存为 DDR3 512 MiB,稳定工作频率为 100 MHz。

本文基于 HDCMPSoC 结构进行改进优化、拓展研究,选择 HDCMPSoC 模拟器^[11]作为本次实验的对比对象。HDCMPSoC 模拟器集成的运算节点包括 8 个 RCU、8 个 GFU 和 8 个 FFT。本文实验中 CAHSCS 网络规模为 8×8,硬件部分携带 FFT、RCU 和 GFP 各 1 个作为功能加速节点,其硬件资源消耗情况见表 2 所列。

表 2 CAHSCS 硬件资源消耗情况

资源类型	使用数量	可使用数量	使用率/%
LUT	4 344	17 600	24.68
LUTRAM	615	6 000	10.25
FF	6 152	35 200	17.48
BRAM	3.5	60.0	5.83
DSP	18	80	22.50
BUFG	1	32	3.13

2.2 测试与性能分析

1) 软硬件环境独立测试。对本文 CAHSCS 软件模拟器执行不同拓扑网络结构测试。本文软件模拟器可支持的拓扑结构包括 2D-Mesh、2D-Tours 和 3D-Cube,在通信注入率为 33%时,软件模拟器性能表现见表 3 所列。

表 3 本文软件模拟器对不同拓扑结构的性能表现

指标	2D-Mesh	2D-Tours	3D-Cube
包延迟/ms	12.38	13.28	10.47
吞吐率/%	21.35	21.42	24.69

硬件加速器独立测试采用加载单一运算任务(任务 1)的方法,执行仅涉及单个计算单元的简单任务,评估硬件加速器的加速效果。本文选择 FFT、向量乘法和向量加法等典型数据/计算密集型任务作为基本测试激励,共设置 3 类 4 组运算任务来验证硬件加速器的性能。

实验结果见表 4 所列,可以看出,单任务实验

中本文设计的硬件加速器加速比在 10 倍以上,符合硬件加速的普遍需求。

表 4 2 种硬件加速器执行任务 1 的性能对比

任务类型	操作用时/ μ s		加速比
	本文	文献[12]	
1×256 点 FFT	5.41	69.4	12.83
50×256 点 FFT	412	5 600	13.59
8×64 向量乘法	35	401.6	11.47
8×64 向量加法	10	233.5	23.35

2) 基本组合任务测试。本组实验目标是验证 CAHSCS 软件模拟器和硬件加速器协同工作下执行一定复杂度任务的性能。测试向量(任务 2)执行过程如图 9 所示,硬件加速器对同一组数据依次执行 3 次 FFT、向量乘法、IFFT 组合过程。

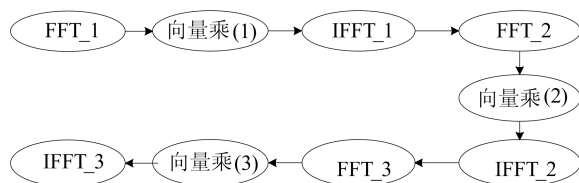


图 9 任务 2 流程图

修改任务 2 原始数据规模,测试不同数据规模下本文 CAHSCS 执行任务 2 的性能变化,同时在 HDCMPSoC^[11]模拟器上完成相同任务,两者实验结果的对比如图 10 所示和见表 5 所列。

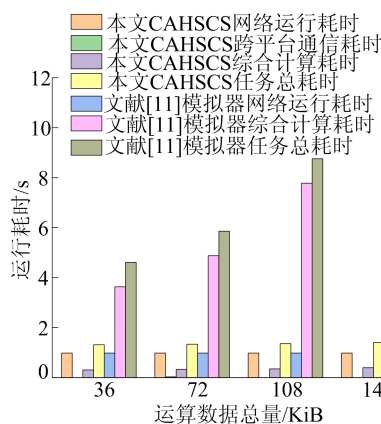


图 10 2 种模拟器执行任务 2 的性能对比

表 5 2 种模拟器执行任务 2 部分操作用时 单位:ms

任务 2	CAHSCS	文献[11]模拟器
网络行为	981	981
跨平台通信	28	
计算总时长	393	9 590
任务总体	1 402	10 571

注:运算数据总量为 144 KiB。

由图 10 可知,CAHSCS 对数据密集和计算密集型任务表现出明显的加速。相较于文献[11]模拟器,CAHSCS 的加速比在 3.51~8.38 之间,且随着运算数据总量的增大而增大,但加速比小于任务 1 的执行结果。

表 5 统计了 CAHSCS 和文献[11]模拟器分别执行相同数据规模的任务 2 时各个操作步骤的用时,得到 CAHSCS 运行加速比,见表 6 所列。由表 6 可知,当 CAHSCS 执行的运算规模有限时,软件模拟器和硬件加速器耗时比约为 7 : 3,软件模拟器消耗 70%的时间,这是造成任务 2 整体加速低于任务 1 的主要原因。

表 6 CAHSCS 执行任务 2 的加速比

运算总量/KiB	36	72	108	144
加速比	3.50	4.40	6.50	7.54

3) 真实任务测试。第 3 个实验目标是研究 CAHSCS 处理密集数据交互任务时的性能特征,测试集合(任务 3)是 SAR 算法中的真实任务。该任务片段包括多次数据交互及多个异构 IP 协同处理,在执行过程中需要频繁执行存储访问和任务同步。任务 3 在 CAHSCS 上的任务映射过程如图 11 所示。

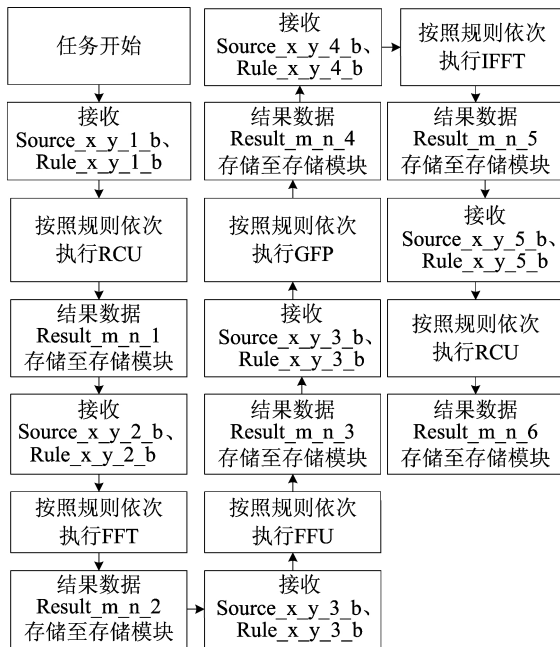


图 11 CAHSCS 任务 3 执行流程

任务 3 实验结果如图 12 所示和见表 7 所列,加速比统计结果见表 8 所列。可以看出,在执行较大规模任务时,硬件加速有明显效果。

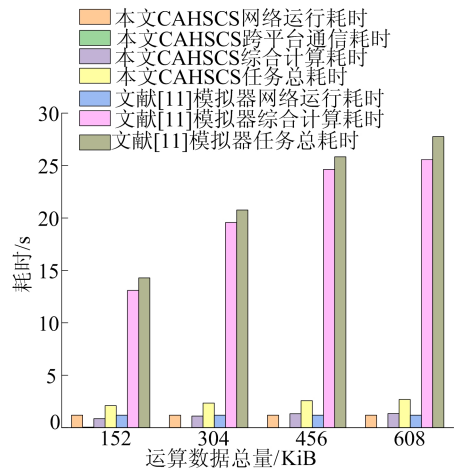


图 12 2 种模拟器执行任务 3 的性能对比

表 7 2 种模拟器执行任务 3 部分操作耗时 单位:ms

任务 3	CAHSCS	文献[11]模拟器
网络运行	1 193	1 193
跨平台通信耗时	52	
综合计算耗时	1 346	25 642
任务总体耗时	2 591	26 835

注:运算数据总量 608 KiB。

表 8 CAHSCS 执行任务 3 的加速比

运算总量/KiB	152	304	456	608
加速比	6.78	8.83	10.03	10.36

综上所述:相较于文献[11]模拟器,CAHSCS 执行组合任务时加速比相对有所降低,而复杂运算的时间占比越大,整体加速比越明显。

3 结 论

本文通过软硬件协同设计方法对片上网络模拟器进行优化设计,并通过实验验证软硬件协同模拟器对片上网络模拟的可行性,探索软硬件协同模拟器对于目标结构进行优化加速的特点。实验结果表明,本文设计的软硬件协同模拟器对目标模拟器的加速具有实际意义,能够提升目标模拟器的运行效率。

[参 考 文 献]

[1] LIT A, LYIA P O, SUHAILI S, et al. Performance evaluation of multi-channel for 10×10 mesh wireless network-on-chip architecture[C]//2022 IEEE International Conference on Computing (ICOCO). [S. l.]:IEEE,2022:150-155.

[2] NOLTING S,PAYÁ-VAYÁ G,GIESEMANN F, et al. Dynamic self-reconfiguration of a MIPS-based soft-core processor architecture[J]. Journal of Parallel and Distributed Computing,2019,133:391-406.

- [3] PÉREZ I, VALLEJO E, MORETO M. BST: a bookSim-based toolset to simulate NoCs with single-and multi-hop bypass[C]//2020 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). [S. l.]: IEEE, 2020: 47-57.
- [4] RAO A, KERNS K. An artificial case to evaluate the scalability bottleneck of a simulator[C]//2022 IEEE 16th International Conference on Solid-State & Integrated Circuit Technology (ICSICT). [S. l.]: IEEE, 2022: 1-3.
- [5] WAIDYSOORIYA H M, OHMA Y, HARIYAMA M. FPGA-based prototype of a quantum annealing simulator for sparse ising model[C]//2022 IEEE 15th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc). [S. l.]: IEEE, 2022: 195-199.
- [6] COMPTON K, HAUCK S. Reconfigurable computing: a survey of systems and software[J]. ACM Computing Surveys (CSUR), 2002, 34(2): 171-210.
- [7] MINAMI D, NAKAMOTO Y, KOGA Y. Simulation environment of embedded control system for multi-core processor with faster CPU simulator [C]//2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics). [S. l.]: IEEE, 2020: 405-410.
- [8] LEFOUL J B, DUGO A T A, MAGALHAES F. Simulator-based framework towards improved cache predictability for multi-core avionic systems [C]//2020 Spring Simulation Conference (SpringSim). [S. l.]: IEEE, 2020: 1-12.
- [9] NAKAMOTO Y, MINAMI D, SHIBA S. A simulation environment of embedded control systems for multi-core processors (WiP report) [C]//2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). [S. l.]: IEEE, 2018: 178-182.
- [10] 刘家华, 陈靖宇. 多核并行脉冲神经网络模拟器的设计[J]. 计算机工程与应用, 2020, 56(22): 244-250.
- [11] 牛云鹏, 张多利, 唐旭, 等. 一种可重构多核系统的仿真加速方案[J]. 微电子学与计算机, 2021, 38(2): 19-24.
- [12] 姚宇, 孙晓霞, 张多利, 等. 一种系统级异构多核模拟器的实现[J]. 合肥工业大学学报(自然科学版), 2018, 41(7): 921-927.
- [13] 王正茂. 面向高密度计算应用的异构多核处理器优化设计[D]. 合肥: 合肥工业大学, 2020.

(责任编辑 胡亚敏)

(上接第 301 页)

- [8] CHHETRY A, SHARMA S, YOON H, et al. Enhanced sensitivity of capacitive pressure and strain sensor based on $\text{CaCu}_3\text{Ti}_4\text{O}_{12}$ wrapped hybrid sponge for wearable applications [J]. Advanced Functional Materials, 2020, 30(31): 1910020.
- [9] SU Y, LI W, CHENG X, et al. High-performance piezoelectric composites via beta phase programming [J]. Nature Communication, 2022, 13(1): 4867.
- [10] KHAN H, MAHMOOD N, ZAVABETI A, et al. Liquid metal-based synthesis of high performance monolayer SnS piezoelectric nanogenerators [J]. Nature Communication, 2020, 11(1): 3449.
- [11] WANG Y, CHEN Z, MEI D, et al. Highly sensitive and flexible tactile sensor with truncated pyramid-shaped porous graphene/silicone rubber composites for human motion detection [J]. Composites Science and Technology, 2022, 217: 109078.
- [12] SHENG N, JI P, ZHANG M, et al. High sensitivity polyurethane-based fiber strain sensor with porous structure via incorporation of bacterial cellulose nanofibers [J]. Advanced Electronic Materials, 2021, 7(4): 2001235.
- [13] WU C, WANG H, LI Y, et al. Sensitivity improvement of stretchable strain sensors by the internal and external structural designs for strain redistribution [J]. ACS Applied Materials and Interfaces, 2020, 12(45): 50803-50811.
- [14] ZHAO W, LIU Y, ZHAO C, et al. A fast self-healable and stretchable conductor based on hierarchical wrinkled structure for flexible electronics [J]. Composites Science and Technology, 2021, 211: 108834.
- [15] HAN Z, LIU L, ZHANG J, et al. High-performance flexible strain sensor with bio-inspired crack arrays [J]. Nanoscale, 2018, 10(32): 15178-15186.
- [16] ZHANG Q, SHEN L, LIU P, et al. Highly sensitive resistance-type flexible pressure sensor for cuffless blood-pressure monitoring by using neural network techniques [J]. Composites Part B (Engineering), 2021, 226: 109365.
- [17] XIA P, LIU P, WU S, et al. Highly stretchable and sensitive flexible resistive strain sensor based on waterborne polyurethane polymer for wearable electronics [J]. Composites Science and Technology, 2022, 221: 109355.
- [18] WU S, LIU P, TONG W, et al. An ultra-sensitive core-sheath fiber strain sensor based on double strain layered structure with cracks and modified MWCNTs/silicone rubber for wearable medical electronics [J]. Composites Science and Technology, 2023, 231: 109816.
- [19] YANG H, LI J, XIAO X, et al. Topographic design in wearable MXene sensors with in-sensor machine learning for full-body avatar reconstruction [J]. Nature Communication, 2022, 13(1): 5311.
- [20] ZHU J, WU X, JAN J, et al. Tuning strain sensor performance via programmed thin-film crack evolution [J]. ACS Applied Materials and Interfaces, 2021, 13(32): 38105-38113.

(责任编辑 胡亚敏)