doi:10.11887/j.cn.202302013 http://journal.nudt.edu.cn

信号处理与深度学习硬件加速的一致性计算结构*

高彦钊[1],陶常勇[2]

(1. 战略支援部队信息工程大学,河南 郑州 450001;2. 天津市滨海新区信息技术创新中心,天津 300450)

摘 要:在计算需求层面对多种典型信号处理算法与深度学习算法进行了分析与模块化分解,提取了两类应用共有的且适合并行硬件加速的计算模块,提出了信号处理与深度学习的一致性计算模型,并基于一致性计算模型设计了控制与计算分离的层次化处理单元与阵列化计算结构。通过对不同应用计算过程的软件定义能够实现信号处理与深度学习的一致性硬件加速计算,基于Zynq计算平台从重构效率与计算性能两个方面对一致性计算模型与计算结构进行了验证,结果表明:基于一致性计算模型的软件定义可重构计算结构,具有较高的计算性能与重构效率。

关键词:深度学习;信号处理;硬件加速;计算结构

中图分类号:TP391 **文献标志码**:A **文章编号**:1001-2486(2023)02-112-09

Hardware-accelerated consistent computing structure for signal processing and deep learning

GAO Yanzhao[1], *TAO Changyong*[2]

(1. Strategic Support Force Information Engineering University, Zhengzhou 450001, China;
2. Information technology Innovation Center of Tianjin Binhai New Area, Tianjin 300450, China)

Abstract: A variety of typical signal processing algorithms and deep learning algorithms were analyzed and modularized from the calculation requirements level. The computing modules, which were suitable for hardware acceleration parallelly in the two types of applications were extracted. A consistent computing model for signal processing and deep learning was proposed, and a hierarchical processing element and arrayed processing structure were proposed based on the consistent computing model in which the control part and computation part were separated. By the software definition of different application computing processes, the consistent hardware-accelerated computation of signal processing and deep learning could be realized flexibly. Based on Zynq computing platform, the consistency computing model and computing structure were verified from two aspects of reconstruction efficiency and computing performance. The validation results indicate that software-defined reconfigurable computing structures based on consistency computing models have high computational performance and reconstruction efficiency.

Keywords: deep learning; signal processing; hardware acceleration; computing structure

近年来,人工智能技术的飞速发展与广泛应用对现代战争[1-2]、工业范式[3-4]以及日常生活[5-6]产生了深刻的影响,尤其是随着边缘设备与移动终端的广泛使用,对未来计算系统提出了更高的要求。

从应用需求角度看,虽然人工智能计算任务在计算系统中占据的比重越来越大,但是在当前以及未来很长的一段时间内,信号或信息处理等科学计算仍然是计算系统任务的重要组成部分。因此,未来计算系统不仅需要支撑深度学习等人工智能处理任务,而且必须能够承担诸如信号处理等科学计算任务。如:基于深度学习的目标检测任务中,复杂天气状况可能会导致图像模糊,需要通过传统信号处理技术对模糊图像进行去雾和图像增强等预处理,然后再采用人工智能算法进行目标检测;对语音识别任务,为了消除人类发声器官本身和语音信号采集设备所带来的混叠和高次谐波失真等因素的影响,必须通过传统信号处理技术对其进行预加重、分帧、加窗等预处理操作,以保证人工智能语音识别阶段的信号更均匀、平滑。传统信号处理与深度学习在算法与成熟度等层面存在较大差异,两者的研究与应用一直呈相对割裂的状态,因此人工智能计算系统往往缺乏对传统信号处理的支持能力,很难实现端到端

* **收稿日期**:2021-04-08
基金项目:国家科技重大专项核高基资助项目(2016ZX01012101)
作者简介:高彦钊(1984—),男,河北平山人,助理研究员,博士,E-mail:buaagaoyz@sina.com;
陶常勇(通信作者),男,山东莱芜人,高级工程师,硕士,E-mail:tcy@ndsc.com.cn

的全流程处理,对信号处理部分需要添加额外的处理模块。然而,一方面深度学习研究热点如卷积神经网络(convolutional neural networks, CNN)、循环神经网络(recurrent neural network, RNN)等算法的计算中包含大量的、可并行化处理的数值计算;另一方面信号处理不论应用场景为何、处理对象为何、计算方法为何、计算器件为何,其计算方法优化、计算过程管理以及计算资源分配等都在逐步向智能化方向发展。因此,两者之间不仅具有明显的相通之处,而且具有强烈的相互支撑、融合发展的必要性。

从计算需求角度看,在数据量爆炸的信息时代,不论是信号处理还是深度学习,均朝着海量数据的实时处理、计算方法灵活调整、计算功耗有效降低、计算过程智能管控以及计算系统稳健可靠等目标发展。在摩尔定律与 Dennard 缩放定律逐步放缓的历史背景下,单纯依靠工艺水平的提高或者在冯诺依曼计算架构下从单核到众核的扩展已经很难应对上述问题。因此,基于粗粒度可重构计算等新型计算方式实现对计算任务的硬件加速受到了越来越多的关注。

对此,本文针对信号处理与深度学习一体化硬件加速需求,在深入分析多种典型信号处理算法与深度学习算法的基础上,针对两者在同一硬件平台加速的计算需求,提出了两者一致性硬件加速的计算方法,并基于软件定义硬件以及可重构计算技术,设计并分析了硬件加速的一致性计算结构,为信号处理与深度学习两大类应用的一体化硬件加速提供了可行的技术思路。

1　典型信号算法分析

1.1　空时自适应处理方法

空时自适应处理(space-time adaptive processing, STAP)是基于一维空域滤波技术发展而来的,目前已成为信号处理领域的重要研究方向。从相控阵雷达各子阵下行信号开始到恒虚警率(constant false alarm rate, CFAR)检测报告,以 m 个时域维度(m-dimension time-domain, mDT)算法[7]为基础的 STAP 信号处理流程及其主要计算模块如图 1 所示。

1.2　脉冲多普勒处理方法

脉冲多普勒(pulse Doppler, PD)雷达是基于多普勒原理的雷达体制,在距离分辨力、速度分辨力以及杂波抑制等方面具有非常突出的能力,能在强杂波背景中分辨出运动目标[8-9]。PD 处理流程及其主要计算模块如图 2 所示。

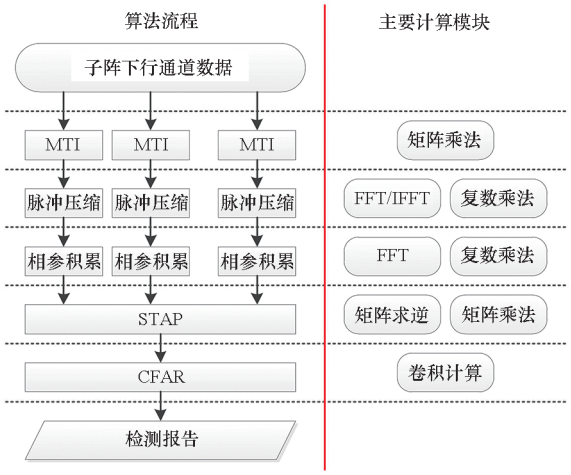

图 1　STAP 算法流程及其主要计算模块

Fig. 1　STAP algorithm flow and its main calculation module

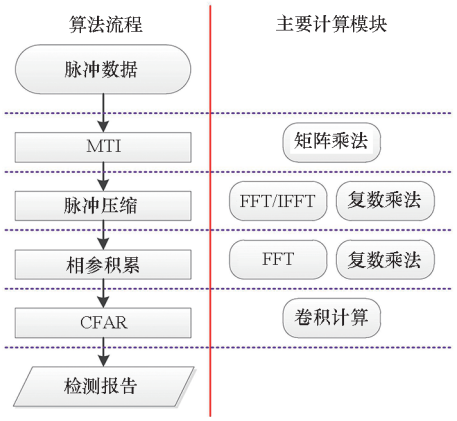

图 2　PD 算法流程及其主要计算模块

Fig. 2　PD algorithm flow and its main calculation module

1.3　大斜视合成孔径雷达成像

合成孔径雷达(synthetic aperture radar, SAR)采用脉冲压缩技术和合成孔径原理实现地面场景全天候、全天时以及远距离成像。与正侧视 SAR 成像相比,大斜视 SAR 成像具有更好的机动性,可通过调整天线指向对感兴趣区域进行多次重复观测[10-11]。大斜视 SAR 成像处理流程及其主要计算模块如图 3 所示。

1.4　遥感光学图像目标识别

对于遥感光学卫星影像中的舰船目标识别问题,为了解决云杂波、海杂波以及舰船浪迹等造成的干扰,克服不同目标尺寸大小对检测带来的困难,文献[12-13]提出了无监督的基于视觉显著性与舰船方向梯度直方图(ship histogram of oriented gradient, S-HOG)描述子的遥感光学图像目标识别算法,其处理流程及主要计算模块如图 4所示。

图 3 大斜视 SAR 成像算法流程及其主要计算模块

Fig. 3 Algorithm flow of high-squint SAR imaging and its main computing module

图 4 遥感光学图像目标识别算法流程及其主要计算模块

Fig. 4 Remote sensing optical image target recognition algorithm flow and its main computing module

2 典型深度学习算法分析

2.1 卷积神经网络

CNN 属于前馈型神经网络，是目前深度学习领域非常具有代表性的神经网络之一，在大型图像处理方面表现出色，目前已广泛应用于图像分类、目标定位等领域。以 LeNet－5[14] 为例，CNN 的处理流程及主要计算模块如图 5所示。

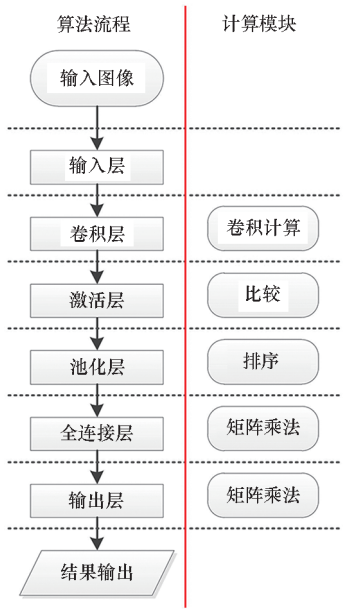

图 5 CNN 算法流程及其主要计算模块

Fig. 5 CNN algorithm flow and its main computing modules

2.2 循环神经网络

RNN 与 CNN 不同，以序列数据作为输入，通过对时序数据进行学习实现上下文信息的存储与表达，具有记忆性与参数共享性，是一种全连接神经网络，已经在自然语言处理领域广泛应用，如语音识别、文本分类和情景分析等。其处理流程与主要计算模块如图 6 所示。

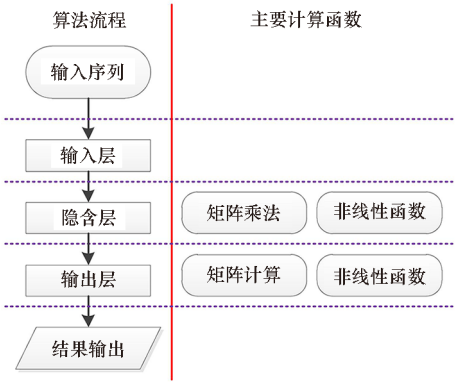

图 6 RNN 算法流程及其主要计算模块

Fig. 6 RNN algorithm flow and its main computing modules

3 一致性计算方法

通过上述对一维脉冲处理、二维脉冲处理、二维 SAR 成像、SAR 图像解译、CNN 以及 RNN 等多个典型算法及其主要计算模块的分析，虽然应用场景不同，计算算法不同，但是其主要计算模块包括 FFT/IFFT、矩阵乘法、矩阵求逆、卷积计算、比

较、排序、复数乘法等。其中,适合基于硬件大规模并行加速计算的模块为 FFT/IFFT、矩阵乘法、矩阵求逆以及卷积计算四类。而事实上,这些计算模块也是信号处理与深度学习硬件加速的主要研究对象[15-20]。

3.1　计算模型

3.1.1　FFT/IFFT

根据 FFT 计算方法,按频率抽取(decimation-in-frequency, DIF)的基-2 蝶形计算表达式为:

$$\begin{cases} Y_1 = \omega_1 X_1 + \omega_2 X_2 \\ Y_2 = \omega_1 X_1 - \omega_2 X_2 \end{cases} \tag{1}$$

同样,按频率抽取的基-4 蝶形计算表达式为:

$$\begin{cases} Y_1 = \omega_1 X_1 + \omega_2 X_2 + \omega_3 X_3 + \omega_4 X_4 \\ Y_2 = \omega_1 X_1 - \omega_2 X_2 - \mathrm{j}\omega_3 X_3 + \mathrm{j}\omega_4 X_4 \\ Y_3 = \omega_1 X_1 + \omega_2 X_2 - \omega_3 X_3 - \omega_4 X_4 \\ Y_4 = \omega_1 X_1 - \omega_2 X_2 + \mathrm{j}\omega_3 X_3 - \mathrm{j}\omega_4 X_4 \end{cases} \tag{2}$$

其中:$Y_i(i=1,2,3,4)$表示蝶形运算计算结果;ω_i $(i=1,2,3,4)$表示蝶形运算的旋转因子;$X_i(i=1,2,3,4)$表示蝶形运算输入。

3.1.2　矩阵乘法

假设矩阵 $\boldsymbol{Y}=\boldsymbol{A}\cdot\boldsymbol{B}$,其中 $\boldsymbol{A}=\{a_{ij}\mid i=1,2,\cdots,M;j=1,2,\cdots,K\}$,$\boldsymbol{B}=\{b_{ij}\mid i=1,2,\cdots,K;j=1,2,\cdots,N\}$,则矩阵 $\boldsymbol{Y}$ 的任一元素 $y_{ij}(i=1,2,\cdots,M;j=1,2,\cdots,N)$表示为:

$$y_{ij} = \sum_{k=1}^{K} a_{ik} b_{kj} \tag{3}$$

3.1.3　矩阵求逆

采用基于 $\boldsymbol{LU}$ 分解的矩阵求逆方法计算矩阵 $\boldsymbol{A}=\{a_{ij}\mid i=1,2,\cdots,N;j=1,2,\cdots,N\}$ 的逆矩阵 $\boldsymbol{Y}=\{y_{ij}\mid i=1,2,\cdots,N;j=1,2,\cdots,N\}$,包括三个步骤:

1)$\boldsymbol{LU}$ 分解,将矩阵 $\boldsymbol{A}$ 分解为上三角矩阵 $\boldsymbol{U}=\{u_{ij}\mid i=1,2,\cdots,N;j=1,2,\cdots,N\}$ 与下三角矩阵 $\boldsymbol{L}=\{l_{ij}\mid i=1,2,\cdots,N;\ j=1,2,\cdots,N\}$,其计算表达式为:

$$u_{ij} = \begin{cases} a_{ij} & i=1;\ j=1,\cdots,N \\ a_{rj} - \sum_{k=1}^{r-1} l_{rk} u_{kj} & r=1,\cdots,N;\ j=r,\cdots,N \end{cases} \tag{4}$$

$$l_{ij} = \begin{cases} a_{ij}/u_{11} & i=1;\ j=1,\cdots,N \\ \dfrac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}}{u_{jj}} & i=j+1,\cdots,N;\ j=1,\cdots,N \end{cases} \tag{5}$$

2)$\boldsymbol{L}$ 与 $\boldsymbol{U}$ 求逆,假设矩阵 $\boldsymbol{L}$ 的逆矩阵表示为 $\boldsymbol{V}=\{v_{ij}\mid i=1,2,\cdots,N;j=1,2,\cdots,N\}$,矩阵 $\boldsymbol{U}$ 的逆矩阵表示为 $\boldsymbol{R}=\{r_{ij}\mid i=1,2,\cdots,N;j=1,2,\cdots,N\}$,其计算表达式分别为:

$$v_{ji} = \begin{cases} l_{ii}^{-1} & i=j \\ -v_{ii}\left(\sum_{k=i+1}^{j} v_{jk} l_{ki}\right) & i<j \\ 0 & i>j \end{cases} \tag{6}$$

$$r_{ij} = \begin{cases} u_{ii}^{-1} & i=j \\ -v_{ii}\left(\sum_{k=i+1}^{j} u_{ik} r_{kj}\right) & i<j \\ 0 & i>j \end{cases} \tag{7}$$

3)$\boldsymbol{L}$ 与 $\boldsymbol{U}$ 乘法,其计算表达式为:

$$y_{ij} = \sum_{k=1}^{N} r_{ik} v_{kj} \quad i=1,\cdots,N;j=1,\cdots,N \tag{8}$$

3.1.4　卷积计算

假设 3×3 维卷积核为 $\boldsymbol{W}=\{w_{ij}\mid i=1,2,3;j=1,2,3\}$,输入图像为 $\boldsymbol{A}=\{a_{ij}\mid i=1,2,\cdots,N;j=1,2,\cdots,N\}$,卷积结果为 $\boldsymbol{Y}=\{y_{ij}\mid i=1,2,\cdots,N-2;j=1,2,\cdots,N-2\}$。则卷积计算结果的任意元素 y_{ij}表示为:

$$\begin{aligned} y_{ij} = {} & w_{11}a_{i-1,j-1} + w_{12}a_{i-1,j} + w_{13}a_{i-1,j+1} + \\ & w_{21}a_{i,j-1} + w_{22}a_{ij} + w_{23}a_{i,j+1} + \\ & w_{31}a_{i+1,j-1} + w_{32}a_{i+1,j} + w_{33}a_{i+1,j+1} \end{aligned} \tag{9}$$

不论 FFT/IFFT、矩阵乘法、矩阵求逆还是卷积计算,如果将其计算输入视为矩阵(其维数可变,且包含一维向量),综合式(1)~(9),上述计算的数学模型可一致性表示为:

$$y_{ij} = \left(\sum a_{ij} b_{ij} + c_{ij}\right) d_{ij} \tag{10}$$

式中,a_{ij}、b_{ij}、c_{ij}、d_{ij}分别是四个计算输入矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$、$\boldsymbol{D}$ 中的元素,y_{ij}为结果矩阵 $\boldsymbol{Y}$ 中的元素。基于式(10),可以一致性描述 FFT/IFFT、矩阵乘法、矩阵求逆以及卷积计算等不同类型计算任务的计算过程。在不同类型计算任务的计算过程中,计算结果 y_{ij}的角标变化规律(表征着计算结果的输出顺序)以及与 y_{ij}的计算相对应的 a_{ij}、b_{ij}、c_{ij}、d_{ij}的角标变化规律是有所不同的,能够根据实际需求进行软件定义。一般来说,基于式(10)的一致性计算公式,各类型计算任务的计算过程主要包括:①根据计算过程设计确定计算结果 y_{ij}角标 i 与 j 的变化规律;②确定实现元素 y_{ij}计算所需要的输入 a_{ij}、b_{ij}、c_{ij}以及 d_{ij}的集合及其地址变化规律;③将所需输入元素集合从存储器中读取出来并组成算式;④通过乘累加模块组成的算粒完成计算过程,并回传 y_{ij}的计算结果。

3.2 计算结构

3.2.1 处理单元结构

虽然 FFT/IFFT、矩阵乘法、矩阵求逆以及卷积计算等不同的计算模块可一致性表示为式(10),但在针对不同计算的具体执行过程中,计算结果的跳变顺序及其对应的计算输入组成方式均有不同。对此,基于式(10)设计的控制与计算分离的层次化软件定义可重构处理单元(processing element, PE)结构设计如图 7 所示。

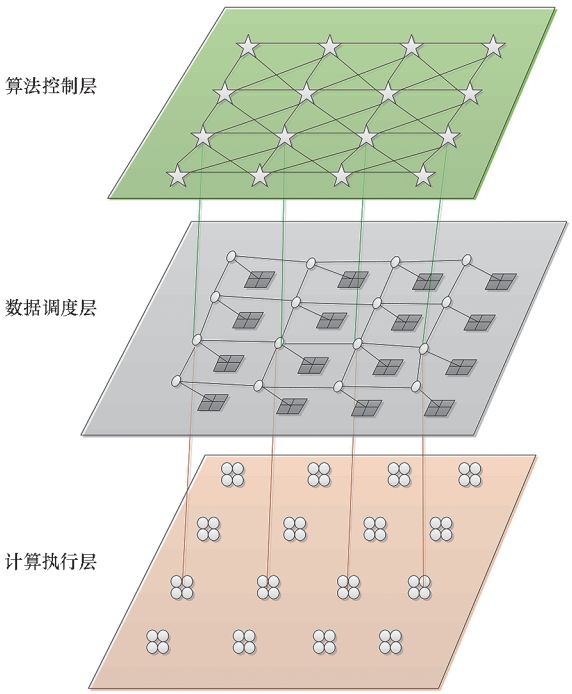

图 7 计算结构示意

Fig. 7 Schematic diagram of calculation structure

在逻辑上 PE 共分为三层,自上而下依次为算法控制层、数据调度层以及计算执行层。其中:算法控制层由多个算式规则控制模块(表示为五角星)组成,每个规则控制模块以软件定义的方式实现对不同计算功能的过程控制,解决“怎么算”的问题,即通过计算结果 $\boldsymbol{Y}$ 的跳变顺序实现计算进程的控制;数据调度层由多块随机存取存储器(random access memory, RAM)组成的分布式数据存储空间(以田字格表示)与算式生成模块(由小圆圈表示)组成,计算数据按不同的存储方式分散存储在多个 RAM 中,并可在层内进行灵活调度,而算式生成模块响应上一层的控制流信息,完成数据的读写访问,解决“算什么”的问题,即根据 $\boldsymbol{Y}$ 的跳变顺序实现计算输入 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$、$\boldsymbol{D}$ 对应元素的选择,并完成计算数据读取;计算执行层由多组乘法器、加法器、累加器、比较器组成,接收待计算数据进行计算并返回计算结果,解决“具体算”的问题,即根据计算输入 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$、$\boldsymbol{D}$ 元素选择执行具体的计算。

同层内各模块之间可以进行信号或数据交互,如:算法控制层各算式规则控制模块之间可以进行控制信号交互,数据调度层各算式生成模块可以读取各个 RAM 的数据,计算执行层内相同的计算模块可以共同完成同一个算式的计算任务等。层间不同模块之间也可实现灵活连接,如:算式规则控制模块可与数据调度层相应位置及其周围的算式生成模块相连接,算式生成模块可与计算执行层相应位置及其周围的计算模块相连接。在计算过程中,配置流先于数据流下发,完成对计算结构的配置,包括模块功能、数据存取以及模块互连等,适应不同的计算任务。

1)算法控制层。不同计算任务可采用不同计算跳转顺序与数据组织形式完成。在计算跳转顺序方面,将算法控制过程分为两个层次:算式间循环控制与算式内循环控制,如图 8 所示。算式间循环控制是第一层循环,指示计算结果 $\boldsymbol{Y}$ 的元素的角标跳转顺序,即计算过程的推进顺序,可以有多种安排方式,根据计算需求而设定;算式内循环控制是第二层循环,指示与当前 $\boldsymbol{Y}$ 计算对应的计算输入 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$、$\boldsymbol{D}$ 的元素的角标解析。在两层循环控制下,不仅可实现不同计算任务的计算顺序控制,而且可快速实现不同计算阶段待计算数据的解析、输入以及存取等,保证计算效率的提升。在数据组织形式方面,根据不同算式间的数据是否可复用将四种应用的计算算式分为两类:组合算式与非组合算式。其中组合算式包括 FFT/IFFT、矩阵乘法与卷积计算,其特点是相邻算式间的计算输入数据可复用,一次数据读取可用于多个算式的计算;非组合算式包括矩阵求逆,其特点是相邻算式间的计算输入数据不可复用,一次数据读取仅用于当前的计算。在组合算式中,充分利用数据复用特性可有效减少数据存取。

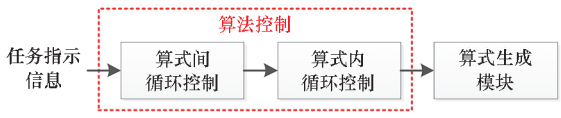

图 8 算法控制模块

Fig. 8 Algorithm control module

2)数据调度层。数据调度接收并解析上层指令,完成数据读取、组合与下发,其功能包括:①将待计算数据按计算需求的方式进行分布式存储,包括按矩阵行列存储、按上下三角矩阵分别存

储、按矩阵元素奇偶分别存储等方式;②算式生成模块按照算式规则控制模块指示实现从 RAM 阵列中任意 RAM 中读取相关待计算数据;③算式生成模块将所读取的数据组成算式并下发至计算执行层;④根据算式规则控制模块指示将计算执行层返回的计算结果按一定的方式存入相应的 RAM 阵列中。其结构示意如图 9 所示,包括 RAM 阵列与算式生成模块,数字用来标识各自的位置。对 RAM 阵列与算式生成模块的索引格式为二元组(i,j),分别表示其行列号。则算式生成模块与 RAM 阵列的位置号是统一的,便于通过配置信息指定相应的路由选择策略,并能够实现任意一个算式生成模块从任意 RAM 中进行数据存取。

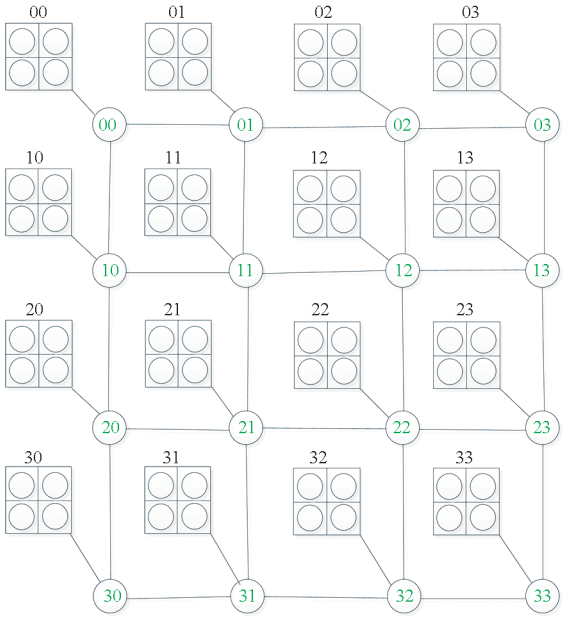

图 9　数据调度层组成示意

Fig. 9　Schematic diagram of data scheduling layer

3)计算执行层。根据式(10),计算执行层中计算模块必须包含多个复数乘法器、复数加法器、复数累加器等基本单元,其中复数乘法器包含四个实数乘法器与两个实数加法器,复数加法器包含两个实数加法器。根据计算场景不同,复数乘法器与复数加法器既可以实现复数乘加运算,也可根据配置信息拆分进行多个实数乘加运算。另外,多个复数乘加运算模块既可单个依次完成一个算式的计算,也可多个并行共同完成一个算式的计算。计算结果通过互连结构返回上层算式生成模块,并根据算法控制模块的指令要求存入相应的 RAM 阵列中。

3.2.2　阵列结构

实现硬件加速的关键在于提高计算能力和数据传输速度[21-22],因此,除了 PE 本身的设计外,由 PE 组成阵列化计算结构实现数据高效传输非常重要。从 PE 的角度看整个计算架构,以 3×3个 PE 组成计算阵列为例,内部由 PE 阵列及数据通路与配置通路组成,并通过串行 RapidIO(serial RapidIO, SRIO)接口、双倍速率同步动态随机存储器(double data rate synchronous dynamic random access memory, DDR SDRAM)(简称 DDR)接口以及本地管理接口与外部连接,其具体组成与互联结构如图 10 所示。阵列结构各模块功能如下:

1)PE 模块:用于完成不同的计算任务,主要包括算法控制层模块、数据调度层模块以及计算执行层模块三大部分。

2)PE 状态控制模块:用于对阵列中的各个 PE 状态进行控制实现多 PE 之间的工作协同,主要包括 PE 接口配置、PE 状态控制(包括空闲、启动、工作及结束等)、PE 间数据流向控制等。

3)Localbus 转 AXI_lite 模块:完成外部软件定义配置或控制命令格式向本地总线格式的转化。

4)AXI_crossbar 模块:实现软件定义配置向各个 PE、DMA0、DMA1 等模块的路由。

5)网络接口模块:自定义 PE 接口与互联网络技术(未在图中标识),实现数据在 PE 之间的路由传输。

6)DMA0 模块:实现 PE 阵列与外部 SRIO 接口之间的数据传输。

7)DMA1 模块:实现 PE 阵列与外部 DDR 接口之间的数据传输。

8)封解包模块:实现 DMA0 与 SRIO 接口之间的数据组帧与切帧。

9)SRIO 接口:实现 PE 阵列数据与片外或板间的数据交互。

10)DDR 接口:实现 DDR 集中式大数据存储与 PE 阵列之间的数据交互。

11)本地管理接口:实现本地软件定义配置或控制指令下发。

通过控制与计算分离的层次化 PE 设计、分布式存储结构设计以及柔性可定义互连结构设计,可实现数据位宽可定义(64 bit 或 32 bit)、PE 功能可定义(FFT、矩阵乘法、矩阵求逆、卷积计算)、数据通道可定义(PE 之间全互联、数据流程可规划)以及计算模式可定义(阵列分割支持时空域计算)等多尺度灵活可重构、兼顾灵活性与高效性的优势。

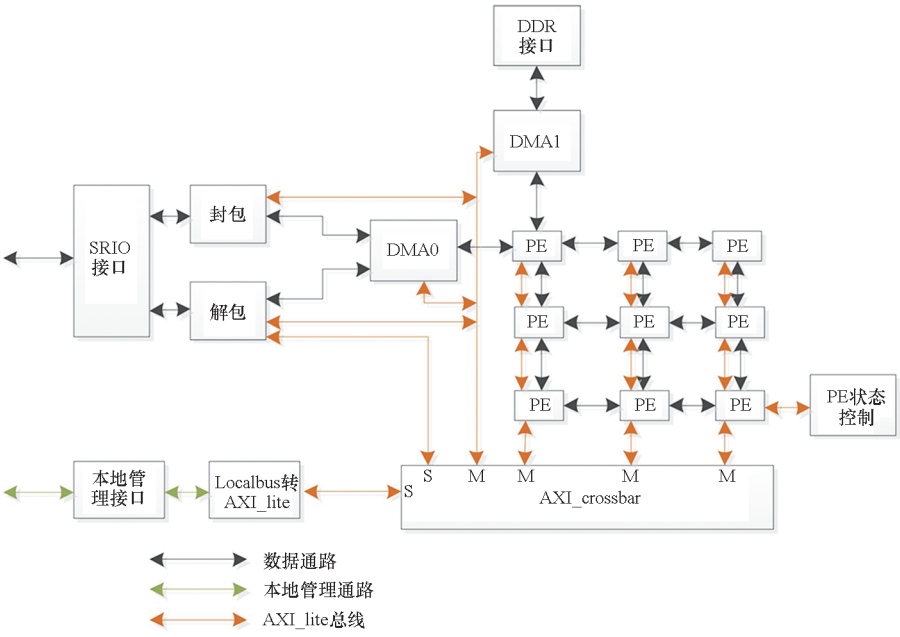

图 10 由 PE 组成的计算阵列

Fig. 10 Computing array composed of PE

4 实验验证

4.1 实验环境

基于本文计算结构的信号处理与深度学习硬件加速的验证实验基于 Xilinx 的 Zynq 开发板(型号为 ZC706)开展,验证环境结构如图 11 所示,其中设计计算阵列 PE 数量为 2 × 3 个。PC 机通过 JTAG 加载 Zynq 逻辑文件,并通过 RART 和以太网接口与 Zynq 上的 ARM 核进行通信。Zynq 的 PS 外挂 DDR、Flash 和以太网 PHY,PS 的 ARM 内核工作频率为 667 MHz,DDR 接口工作频率为 533 MHz,计算阵列工作频率为 100 MHz,计算精度为单精度浮点,复数数据宽度为 64 bit,实数数据宽度为 32 bit。

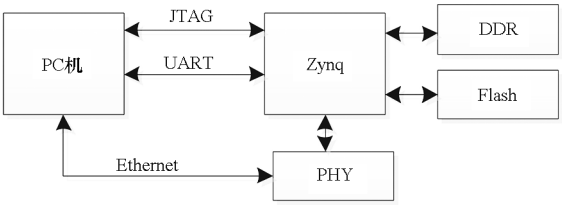

图 11 验证环境结构

Fig. 11 Verification environment structure

4.2 实验结果

4.2.1 实验一:计算阵列重构效率

单个 PE 配置文件为 25.6 Kbit,全阵列 6 个 PE 配置文件为 153.6 Kbit,包括算法控制层中算式间循环控制配置与算式内循环控制配置、数据调度层中数据存储方式配置与数据读取方式配置、计算执行层中对计算模块的配置以及计算阵列中数据传输路径的配置四个部分。在配置数据通路位宽为 32 bit、时钟频率为 50 MHz 的条件下,实现单个 PE 的配置耗时 19.2 μs,实现全阵列 6 个 PE 的配置耗时 115.2 μs,与 FPGA 秒级的 bit 文件加载时间相比,具有巨大的重构效率优势。

4.2.2 实验二:FFT 计算性能

将本文计算结构实现 1 K 点 FFT 的计算性能与其他处理器进行对比,包括 RASP[23] 可重构处理器 NoC[24]、MorphoSys[25] 以及 TI 公司 C6678 等。因各类处理器的工作时钟频率不同,本文所提计算结构的工作频率仅为 100 MHz,为方便比较,将本文方法的计算时间按时钟频率为 1 GHz 进行等比例折算。考虑到 FFT 计算对数据分布无要求,主要考量计算结果的正确性,因此计算数据集随机生成,数据精度为单精度浮点,数据位宽为 64 bit(实部虚部各 32 bit)。各类处理器的 FFT 计算性能对比见表 1。

表 1 FFT 计算时间对比

Tab. 1 Comparison of calculation time of FFT

处理器	RASP	NoC	MorphoSys	C6678	本文方法
计算时间/μs	2.57	76.30	7.40	12.50	1.26

从表 1 中可以看出,基于本文一致性计算方

法及可重构计算结构实现 1 K 点单精度浮点 FFT 计算仅需 1.26 μs,计算性能是 RASP 的 2.04 倍,是 NoC 的 60.56 倍。

4.2.3　实验三:矩阵乘法计算性能

将本文计算结构实现单精度浮点实数的两个 128 × 128 维矩阵相乘的计算性能与其他基于 FPGA 的矩阵乘法器进行对比。考虑到矩阵乘法计算对数据分布无要求,主要考量计算结果的正确性,因此随机生成计算数据集,数据精度为单精度浮点,数据位宽为 32 bit。与文献中各硬件加速结构计算性能的对比见表 2。从表中可以看出,基于本文一致性计算结构实现矩阵乘法计算在同工作时钟频率下优于基于 FPGA 的矩阵乘法计算性能。

表 2　矩阵乘法计算时间对比

Tab.2　Comparison of calculation time of matrix multiplication

计算性能	方法一[26]	方法二[27]	本文方法
矩阵维数	100 × 100	128 × 128	128 × 128
时钟频率/MHz	60	250	100
计算时间/μs	1 351	986.30	908.76

4.2.4　实验四:矩阵求逆计算性能

将本文计算结构实现单精度浮点实数的32 × 32 维矩阵求逆的计算性能与其他基于 FPGA 的矩阵求逆计算器进行对比。计算数据集随机生成,数据精度为单精度浮点,数据位宽为 32 bit。与文献中各硬件加速结构计算性能对比见表 3。在相同工作时钟频率下,针对相同维数的矩阵求逆计算本文方法优于其他基于 FPGA 的矩阵求逆计算性能。

表 3　矩阵求逆计算时间对比

Tab.3　Comparison of calculation time of matrix inverse

计算性能	方法一[28]	方法二[29]	方法三[30]	本文方法
矩阵维数	32 × 32	32 × 32	32 × 32	32 × 32
时钟频率/MHz	100	100	100	100
计算时间/μs	70.81	53.82	87.34	48.36

5　结论

本文在对信号处理与深度学习典型算法分析的基础上,提取了两类应用共有且适合并行加速的计算模块,提出了信号处理与深度学习的一致性硬件加速计算模型并设计了控制与计算分离的层次化软件定义可重构计算结构,在该结构中通过 PE 内算法控制、数据调度以及计算执行等层次化设计、分布式存储结构设计以及 PE 间软件定义互连设计,能够实现 PE 内与 PE 间多尺度灵活重构,不仅可以满足信号处理与深度学习典型计算算法的一体化硬件加速需求,而且基于 FFT、矩阵乘法与矩阵求逆等模块从重构效率和计算性能两个方面与多类硬件加速结构进行了对比,实验验证结果表明,该计算结构具有较高的灵活性与计算性能。

参考文献(References)

[1]　孙强. 人工智能对现代战争的影响[J]. 数码世界, 2018(5): 446.
SUN Q. The influence of artificial intelligence on modern warfare[J]. Digital Space, 2018(5): 446. (in Chinese)

[2]　陆震. 人工智能在军用机器人的应用[J]. 兵器装备工程学报, 2019, 40(5): 1 – 5.
LU Z. Military robots and AI[J]. Journal of Ordnance Equipment Engineering, 2019, 40(5): 1 – 5. (in Chinese)

[3]　胡冰洋. 推动我国第四次工业革命及颠覆性技术创新的分析和建议[J]. 中国经贸导刊, 2019(15): 30 – 33.
HU B Y. Analysis and suggestions on promoting the fourth industrial revolution and subversive technological innovation in China[J]. China Economic & Trade Herald, 2019(15): 30 – 33. (in Chinese)

[4]　薛加玉. 人工智能赋能制造业转型升级[J]. 现代工业经济和信息化, 2019, 9(3): 9 – 10, 16.
XUE J Y. Transformation and upgrading of AI enabling manufacturing industry[J]. Modern Industrial Economy and Informationization, 2019, 9(3): 9 – 10, 16. (in Chinese)

[5]　曾伟良, 吴淼森, 孙为军, 等. 自动驾驶出租车调度系统研究综述[J]. 计算机科学, 2020, 47(5): 181 – 189.
ZENG W L, WU M S, SUN W J, et al. Comprehensive review of autonomous taxi dispatching systems[J]. Computer Science, 2020, 47(5): 181 – 189. (in Chinese)

[6]　谢林利. 智慧城市中基于异构物联网的智慧家居[J]. 计算机科学与应用, 2020(1): 29 – 34.
XIE L L. Smart home based on heterogeneous internet of things in smart city[J]. Computer Science and Application, 2020(1): 29 – 34. (in Chinese)

[7]　向聪, 冯大政, 和洁. 机载雷达三维空时两级降维自适应处理[J]. 电子与信息学报, 2010, 32(8): 1869 – 1873.
XIANG C, FENG D Z, HE J. Three-dimensional spatial-temporal two-step dimension-reduced adaptive processing for airborne radar[J]. Journal of Electronics & Information Technology, 2010, 32(8): 1869 – 1873. (in Chinese)

[8]　袁兴生, 段红, 姚新宇, 等. 脉冲多普勒雷达信号处理仿真系统研究[J]. 计算机应用, 2009, 29(增刊 2): 294 – 296, 300.
YUAN X S, DUAN H, YAO X Y, et al. Study of signal processing simulation system of PD radar[J]. Journal of Computer Applications, 2009, 29(Suppl 2): 294 – 296, 300. (in Chinese)

[9]　姚旺, 金红新, 赵鹏飞, 等. 基于多 DSP 的 PD 脉冲压缩雷达信号处理机的设计[J]. 电子技术应用, 2017, 43(7): 51 – 54.
YAO W, JIN H X, ZHAO P F, et al. Design of PD radar signal processor based on multi-DSP[J]. Application of

Electronic Technique, 2017, 43(7): 51 – 54. (in Chinese)

[10] 顾福飞, 张群, 杨秋, 等. 基于NCS算子的大斜视SAR压缩感知成像方法[J]. 雷达学报, 2016, 5(1): 16 – 24.
GU F F, ZHANG Q, YANG Q, et al. Compressed sensing imaging algorithm for high-squint SAR based on NCS operator[J]. Journal of Radars, 2016, 5(1): 16 – 24. (in Chinese)

[11] 李震宇, 陈溅来, 梁毅, 等. 带有多普勒中心空变校正的大斜视SAR成像方法[J]. 西安电子科技大学学报, 2016, 43(3): 19 – 24.
LI Z Y, CHEN J L, LIANG Y, et al. Imaging method for highly squinted SAR with spatially-variant Doppler centroid correction[J]. Journal of Xidian University, 2016, 43(3): 19 – 24. (in Chinese)

[12] 漆昇翔. 视觉显著性及其在自动目标识别系统中的应用[D]. 武汉: 华中科技大学, 2015.
QI S X. Visual saliency detection with its applications in automatic target recognition systems[D]. Wuhan: Huazhong University of Science and Technology, 2015. (in Chinese)

[13] 白婷. 基于视觉显著性的红外小目标检测算法研究[D]. 武汉: 华中科技大学, 2016.
BAI T. Research algorithms of infrared small target detection based on visual saliency[D]. Wuhan: Huazhong University of Science and Technology, 2016. (in Chinese)

[14] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278 – 2324.

[15] 龚彤艳, 张广婷, 贾海鹏, 等. 一种偶数基Cooley-Tukey FFT高性能实现方法[J]. 计算机科学, 2020, 47(1): 31 – 39.
GONG T Y, ZHANG G T, JIA H P, et al. High-performance implementation method for even basis of Cooley-Tukey FFT[J]. Computer Science, 2020, 47(1): 31 – 39. (in Chinese)

[16] 张多利, 张玲佳, 宋宇鲲. 变维度FFT硬件加速器结构设计及FPGA实现[J]. 微电子学与计算机, 2017, 34(12): 34 – 39, 44.
ZHANG D L, ZHANG L J, SONG Y K. Structure design and FPGA implementation of a variable dimension FFT hardware accelerator [J]. Microelectronics & Computer, 2017, 34(12): 34 – 39, 44. (in Chinese)

[17] 杨飞, 马昱春, 侯金, 等. 基于MPSoC并行调度的矩阵乘法加速算法研究[J]. 计算机科学, 2017, 44(8): 36 – 41.
YANG F, MA Y C, HOU J, et al. Research on acceleration of matrix multiplication based on parallel scheduling on MPSoC[J]. Computer Science, 2017, 44(8): 36 – 41. (in Chinese)

[18] 于敬巨, 张多利, 宋宇鲲. 高性能矩阵求逆硬件加速器的设计与实现[J]. 合肥工业大学学报(自然科学版), 2018, 41(12): 1652 – 1658.
YU J J, ZHANG D L, SONG Y K. Design and implementation of high performance matrix inverse hardware accelerator[J]. Journal of Hefei University of Technology (Natural Science), 2018, 41(12): 1652 – 1658. (in Chinese)

[19] 杨博文, 杨海涛, 高浩浩. CNN加速器中卷积计算单元的硬件设计[J]. 数字技术与应用, 2019, 37(10): 136 – 137.
YANG B W, YANG H T, GAO H H. Hardware design of convolutional computing unit in CNN accelerator[J]. Digital Technology & Application, 2019, 37(10): 136 – 137. (in Chinese)

[20] 秦华标, 曹钦平. 基于FPGA的卷积神经网络硬件加速器设计[J]. 电子与信息学报, 2019, 41(11): 2599 – 2605.
QIN H B, CAO Q P. Design of convolutional neural networks hardware acceleration based on FPGA [J]. Journal of Electronics & Information Technology, 2019, 41(11): 2599 – 2605. (in Chinese)

[21] AHMAD A, PASHA M A. Optimizing hardware accelerated general matrix-matrix multiplication for CNNs on FPGAs[J]. IEEE Transactions on Circuits and Systems Ⅱ: Express Briefs, 2020, 67(11): 2692 – 2696.

[22] KALA S, NALESH S. Efficient CNN accelerator on FPGA[J]. IETE Journal of Research, 2020, 66(6): 733 – 740.

[23] 何国强, 李丽, 李世平. 面向雷达信号处理应用的可重构处理器设计[J]. 现代雷达, 2016, 38(8): 46 – 50, 87.
HE G Q, LI L, LI S P. Design of reconfigurable processor for radar signal processing application[J]. Modern Radar, 2016, 38(8): 46 – 50, 87. (in Chinese)

[24] BAHN J H, YANG J S, HU W H, et al. Parallel FFT algorithms on network-on-chips [J]. Journal of Circuits System & Computers, 2009, 18(2): 255 – 269.

[25] KAMALIZAD A H, PAN C, BAGHERZADEH N, et al. Fast parallel FFT on a reconfigurable computation platform[C]// Proceedings of the 15th Symposium on Computer Architecture and High Performance Computing, 2003.

[26] 田翔, 周凡, 陈耀武, 等. 基于FPGA的实时双精度浮点矩阵乘法器设计[J]. 浙江大学学报(工学版), 2008, 42(9): 1611 – 1615.
TIAN X, ZHOU F, CHEN Y W, et al. Design of field programmable gate array based real-time double-precision floating-point matrix multiplier [J]. Journal of Zhejiang University (Engineering Science), 2008, 42(9): 1611 – 1615. (in Chinese)

[27] 刘沛华, 鲁华祥, 龚国良, 等. 基于FPGA的全流水双精度浮点矩阵乘法器设计[J]. 智能系统学报, 2012, 7(4): 302 – 306.
LIU P H, LU H X, GONG G L, et al. Design of an FPGA-based double-precision floating-point matrix multiplier with pipeline architecture[J]. CAAI Transactions on Intelligent Systems, 2012, 7(4): 302 – 306. (in Chinese)

[28] 陈晓东, 李世平, 何国强. 基于FPGA的Cholesky分解矩阵求逆[J]. 现代雷达, 2019, 41(10): 58 – 61, 67.
CHEN X D, LI S P, HE G Q. Matrix inversion with Cholesky decomposition based on FPGA[J]. Modern Radar, 2019, 41(10): 58 – 61, 67. (in Chinese)

[29] 张多利, 蒋雯, 叶紫燕, 等. 一种用于矩阵求逆的原位替换算法及硬件实现[J]. 合肥工业大学学报(自然科学版), 2020, 43(1): 75 – 80.
ZHANG D L, JIANG W, YE Z Y, et al. An in situ substitution algorithm for matrix inversion and its hardware implementation[J]. Journal of Hefei University of Technology (Natural Science), 2020, 43(1): 75 – 80. (in Chinese)

[30] 张多利, 叶紫燕, 邱俊豪, 等. 任意阶矩阵求逆的算法优化和硬件实现[J]. 合肥工业大学学报(自然科学版), 2019, 42(9): 1227 – 1233.
ZHANG D L, YE Z Y, QIU J H, et al. Optimization of arbitrary order matrix inversion algorithm and its implementation on FPGA[J]. Journal of Hefei University of Technology (Natural Science), 2019, 42(9): 1227 – 1233. (in Chinese)