



(12) 发明专利

(10) 授权公告号 CN 114138342 B

(45) 授权公告日 2022.04.26

(21) 申请号 202210116933.1

(22) 申请日 2022.01.25

(65) 同一申请的已公布的文献号
申请公布号 CN 114138342 A

(43) 申请公布日 2022.03.04

(73) 专利权人 北京大学
地址 100871 北京市海淀区颐和园路5号

(72) 发明人 付晓霞 严伟 罗国杰 郭一江
时广轶 石弼钊

(74) 专利代理机构 北京万象新悦知识产权代理
有限公司 11360

代理人 黄凤茹

(51) Int. Cl.

G06F 9/38 (2006.01)

G06F 13/12 (2006.01)

G06F 15/17 (2006.01)

(56) 对比文件

CN 111078287 A, 2020.04.28

CN 110858387 A, 2020.03.03

CN 111126583 A, 2020.05.08

CN 112445526 A, 2021.03.05

CN 111190840 A, 2020.05.22

CN 112130901 A, 2020.12.25

CN 103926928 A, 2014.07.16

严伟. 对PS0结构的设计与优化.《电子制
作》.2016, (第02期),

雷思磊. 开源处理器Rocket的自定义指令研
究与测试.《单片机与嵌入式系统应用》.2017, 第
17卷(第05期),

A. De.Hardware Assisted Buffer
Protection Mechanisms for Embedded RISC-
V.《in IEEE Transactions on Computer-Aided
Design of Integrated Circuits and
Systems》.2020,

审查员 邱祥吉

权利要求书3页 说明书10页 附图6页

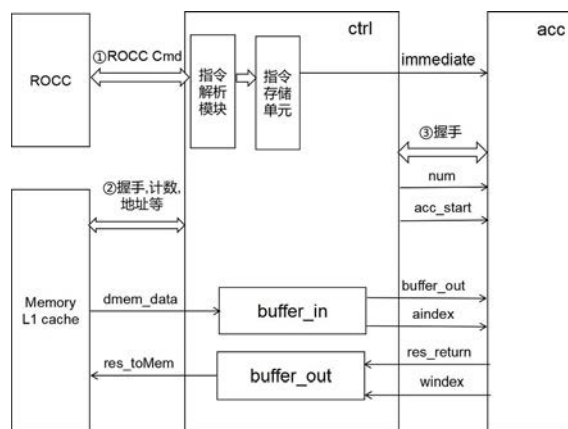
(54) 发明名称

ROCC协处理器接口模型及其自动生成工具
和实现方法

(57) 摘要

本发明公布了一种ROCC协处理器接口模型
及其自动生成工具和实现方法,包括:指令解析
模块、指令存储单元、输入数据存储器、计算结果
存储器、读入数据状态机模块和接口行为状态机
模块;指令解析模块与ROCC协处理器端相连接;
输入数据存储器和计算结果存储器均与内存L1
cache端相连。自动生成工具包括:接口生成功能
实现和接口生成主函数;首先生成ROCC协处理器
接口文件和用于用户测试使用的指令头文件,再
将ROCC协处理器集成到RISC-V系统,实现ROCC协
处理器接口模型的生成。采用本发明技术方案,
可简化和屏蔽RISC-V指令和ROCC接口的具体实
现细节,快速生成得到适配协处理器的硬件接
口。

CN 114138342 B



1. 一种ROCC协处理器接口系统,其特征是,包括:指令解析模块、指令存储单元、输入数据存储器buffer_in、计算结果存储器buffer_out、读入数据状态机m_state和接口行为状态机s_state;指令解析模块与ROCC端相连接;输入数据存储器buffer_in和计算结果存储器buffer_out均与所述ROCC协处理器接口系统之外的内存L1 cache端相连;其中:

A. 所述指令解析模块用于接收从ROCC端传递来的指令数据;通过判断指令数据的类型序号,将指令数据分类;并将指令数据相关信息存放在对应的指令存储单元;指令数据相关信息包括每组指令数据的地址和长度;

B. 所述输入数据存储器buffer_in用于在所述接口行为状态机s_state和所述读入数据状态机m_state的协同作用下,与L1 cache端建立握手协议,接收L1 cache端发来的数据并保存;当输入数据存储器中的数据存满后,发送给ROCC协处理器,可对输入数据存储器的位宽和深度进行调整;

C. 所述计算结果存储器buffer_out用于在接口行为状态机作用下,接收ROCC协处理器传递来的计算结果数据并保存;当全部传递完成后,通过与L1 cache端建立握手协议,将结果数据发送给L1 cache端;

D. 所述读入数据状态机m_state用于调度输入数据从L1 cache端到输入数据存储器,可通过对所述指令解析模块得到的指令存储单元的遍历访问,切换L1 cache端的数据地址,并根据每组数据长度信息,实现多组数据的传输;

E. 所述接口行为状态机s_state用于通过定义多个状态实现协处理器的数据搬运和协处理器驱动。

2. 如权利要求1所述的ROCC协处理器接口系统,其特征是,所述指令解析模块中,通过判断从ROCC协处理器传递进来的指令数据的类型序号,将指令数据分类;类别包括:传递数据的组数、配置协处理器数据、计算数据、结果返回地址、直接传递指令。

3. 如权利要求1所述的ROCC协处理器接口系统,其特征是,所述计算结果存储器的位宽和深度可根据协处理器计算结果的大小进行调整。

4. 如权利要求1所述的ROCC协处理器接口系统,其特征是,所述读入数据状态机包括多个状态:m_idle、m_read、m_pad和m_absorb,分别表示:

m_idle:初始空闲状态;

m_read:当输入数据的位宽与输入数据存储器的位宽相同时,从L1 cache端读入到输入数据存储器;

m_pad:当输入数据的位宽小于输入数据存储器的位宽时,从L1 cache端读入输入数据存储器,通过比较输入数据位宽与buffer_in输入数据存储器位宽的差异,进行空缺位补零,并添加数据结束标志位操作;

m_absorb只用于m_pad的下一个状态;与m_pad共同用于传递小于buffer_in位宽数据即一组数据中末端的数据。

5. 如权利要求1所述的ROCC协处理器接口系统,其特征是,所述接口行为状态机包括多个状态:s_idle、s_absorb、s_finish、s_calculate、s_store和s_write;其中:

s_absorb状态:在读入数据状态机m_state的作用下调度读入数据,当buffer_in存储L1 cache端数据存满时,接口行为状态机s_state的状态由s_idle切换到s_absorb,将buffer_in中的数据传递到协处理器,将buffer_in中数据全部传输完成后进入到s_finish

状态进行判断,判断是否所有配置数据是否传递完成,若完成则进入s_calculate状态,若未完成则继续返回到s_idle状态,等待buffer_in数据存满,进行下一轮向下处理器传递数据;

在经过判断配置数据全部传递完成,即由s_finish状态进入到s_calculate状态;

在s_calculate状态时,包括:a.将输入数据从L1 cache端传输到buffer_in,再从buffer_in传输到ROCC协处理器;b.通过协处理器对不断接收的计算数据进行运算操作,直至所有计算数据传递完成;所有计算数据被运算完成后,由协处理器发出完成运算信号,由s_calculate状态跳转到运算结果数据存储状态s_store;

在存储状态s_store时,通过接口模块与ROCC协处理器建立起握手信号,开始进行运算结果数据返回,从协处理器端传递到计算结果存储器buffer_out;

在完成数据存储后,进入到数据写回状态s_write,通过接口与L1 cache端建立握手,运算结果数据由buffer_out传递到所述指令存储单元保存的输出结果地址处。

6.一种如权利要求1所述的ROCC协处理器接口系统的自动生成工具,其特征是,包括:接口生成功能实现inst_gen.hpp和接口生成主函数main.cpp;

所述接口生成主函数main.cpp用于接收用户对接口系统的相关配置参数;所述接口生成功能实现inst_gen.hpp用于定义接口生成功能实现的多个函数;

通过调用所述接口生成功能实现inst_gen.hpp中的函数,实现所述的ROCC协处理器接口系统的生成。

7.一种如权利要求6所述的ROCC协处理器接口系统的自动生成工具的实现方法,包括如下步骤:

1)用户自定义协处理器有关参数;

确定用户自定义协处理器的输入数据和输出数据信息;其中,用户自定义数据包括输入数据指令数data_num、输入数据地址data_addr、输入数据长度data_len、输出结果地址output_addr、直接传递到协处理器端的指令数值direct_value、输入数据存储器buffer_in和计算结果存储器buffer_out大小数值;

2)根据用户自定义协处理器的输入数据,确定指令解析的次数和指令存储单元的大小,并定义输入数据存储器buffer_in和计算结果存储器buffer_out的大小;

3)设计ROCC协处理器接口系统生成的主函数main.cpp和接口生成功能实现函数inst_gen.hpp,通过判断用户输入数据指令中各个字段位置的不同,对用户输入数据指令采用多种方法进行指令解析,获取一组数据的数据地址和数据长度;

4)根据用户自定义的输入数据,确定从内存L1 cache端传送数据进入接口系统的数据组数;再根据指令数据的长度信息,生成不同次数的切换状态;

通过用户给定的输入数据组数data_num值,确定生成的接口模块要从内存L1 cache端传送多少组数据进入接口系统;

根据指令数据的长度信息,进行读入数据的地址和长度信息的切换,根据用户给定的输入数据组数data_num值的不同,在接口系统中依次对data_num组数据进行传送;

5)通过用户给定的输入数据组数num值,定义传送行为的状态,生成接口系统的行为状态机;

将用户自定义的输入数据中的数据组数data_num值作为s_finish状态的状态跳转判

断条件,若现在传递数据组数数值刚好为data_num,进入s_calculate状态;其余状态为固定形式,描述采用字符输出;行为状态机的状态包括:s_idle、s_absorb、s_finish、s_calculate、s_store和s_write;其中在s_finish状态时进行判断,判断是否所有配置数据是否传递完成,若完成则进入s_calculate状态,若未完成则继续返回到s_idle状态;

通过上述步骤2)~5),即可生成ROCC协处理器接口模块文件,实现对加速器开发人员简化和屏蔽ROCC协处理器接口系统;

6)通过对用户自定义的输入数据中的指令名和指令给定的顺序和个数进行识别,将用户给定的简化指令转换成RISC-V自定义ISA,生成指令描述文件;

通过上述步骤,可得到协处理器接口文件和指令描述头文件,即构建得到ROCC协处理器接口系统的自动生成工具;通过利用ROCC协处理器接口系统的自动生成工具,实现ROCC协处理器接口模块的自动生成。

8.如权利要求7所述的ROCC协处理器接口系统的自动生成工具的实现方法,其特征是,步骤3)中,通过判断用户输入数据指令中各个字段位置的不同,对用户输入数据指令采用多种方法进行指令解析,包括如下解析过程:

a.对于从ROCC协处理器获取信息的指令im.create_instr("DATA", "data1_addr", "data1_len"),指令中"DATA", "data1_addr", "data1_len"分别为第一字段、第二字段和第三字段,第一字段为指令名;第二字段和第三字段位置均非空;解析该指令时,从ROCC协处理器获取rs1和rs2两个值;rs1,rs2为RISC-V自定义ISA中的源寄存器;

b.对于从ROCC协处理器获取信息的指令im.create_instr("OUTPUT", "output_addr"),指令中"OUTPUT", "output_addr"分别为第一字段、第二字段;第一字段为指令名;第二字段位置非空;解析该指令时,从ROCC协处理器获取rs1,输出rs1值。

9.如权利要求7所述的ROCC协处理器接口系统的自动生成工具的实现方法,其特征是,用户自定义的输入数据中的指令名包括DATA、OUTPUT或DIRECT_VALUE;可根据用户给出指令的先后次序识别指令的顺序和个数,将用户给定的简化指令转换成RISC-V自定义ISA。

10.如权利要求9所述的ROCC协处理器接口系统的自动生成工具的实现方法,其特征是,步骤6)中,将用户给定的简化指令转换成RISC-V自定义ISA的格式包括:costomx rd, rs1,rs2,func;其中:

用户给定:im.create_instr("DATA", "data1_addr", "data1_len");

生成ISA:#define DATA(arg1, arg2) \

ROCC_INSTRUCTION_DSS(2, 0, arg1, arg2, DATA_ID)

即生成指令描述文件。

ROCC协处理器接口模型及其自动生成工具和实现方法

技术领域

[0001] 本发明涉及RISC-V架构扩展加速器内核的RTL接口技术,尤其涉及一种Rocket定制协处理器(Rocket Custom Coprocessor,ROCC)接口模型、接口模型自动生成方法及工具,包括ROCC扩展指令协议的硬件控制器、软件仿真器、以及带加速器的RISC-V代码生成工程的接口自动生成工具及实现方法。

背景技术

[0002] Chipyard 是一个开源框架,用于敏捷开发基于Chisel生成器的片上系统(System-on-Chip,SoC)。Chipyard 是由加州大学伯克利分校(UCB)伯克利架构研究小组(Berkeley Architecture Research,UCB-BAR)开发。它允许利用Chisel硬件构造语言、Rocket Chip SoC生成器和其他UCB-BAR项目来生成 RISC-V SoC,其中包含从MMIO映射外设到自定义加速器的所有内容。包括处理器内核、加速器、工具链、存储系统以及其他外设和工具。

[0003] RISC-V是一个基于精简指令集(RISC)原则的开源指令集(ISA)架构。在Chipyard框架下,有一些加速器通过ROCC接口实现了与具有ROCC协处理器接口的RISC-V内核Rocket(An in-order RISC-V core)或BOOM(Berkeley Out-of-Order Machine)相连,例如:Gemmini项目是一种基于脉动阵列的矩阵乘法单元生成器,它采用非标准RISC-V自定义指令,接入Rocket或BOOM的ROCC端口的RoCC加速器;Hwacha项目是一个解耦矢量架构协处理器,它目前使用矢量架构编程模型实现了非标准的 RISC-V 拓展;SHA3项目是SHA3哈希算法的固定函数加速器,使用ROCC接口的一些Chipyard集成流。

[0004] 虽然这些项目都成功的集成到Chipyard开源框架,实现了与具有ROCC协处理器接口的RISC-V内核相连,但是每个项目其接口都存在很大差异性。若有开发人员设计一款自定义加速器,为了测试加速器性能,想要与具有ROCC协处理器接口的RISC-V内核Rocket相连,就要设计适用其自定义加速器的接口,开发人员必须要对ROCC接口的特点有详细的了解,但这个过程对于不熟悉RISC-V指令的开发人员难度较大,并且从头设计一个适用自己加速器的硬件接口也大大增加了时间开销。

发明内容

[0005] 为了克服上述现有技术的不足,本发明提供ROCC协处理器接口模型及其自动生成工具和实现方法,目的是为定制化ROCC协处理器提供抽象化接口模型,可以对加速器开发人员简化和屏蔽RISC-V指令和ROCC接口的具体实现细节,应用ROCC协处理器接口模型自动生成工具快速得到适配该协处理器的硬件接口。

[0006] 本发明的原理是:设计一种加速器(ROCC协处理器)的抽象接口模型,满足常见的几种加速器的接口需求,并根据用户给出的加速器设计确定待处理数据的组数、地址和长度信息以及结果地址等信息,用ROCC协处理器接口模型自动生成工具生成该加速器的抽象接口文件和指令头文件,将文件分别放到对应硬件设计和测试程序处,执行硬件仿真测试,

实现可定制加速器集成到RISC-V系统框架。

[0007] 本发明提供的技术方案是：

[0008] 一种ROCC协处理器接口模型，包括：指令解析模块、指令存储单元、输入数据存储器buffer_in、计算结果存储器buffer_out、读入数据状态机模块m_state和接口行为状态机模块s_state；指令解析模块与ROCC协处理器端相连接；输入数据存储器和计算结果存储器均与ROCC协处理器接口模型之外的内存L1 cache端相连；其中：

[0009] A. 指令解析模块用于接收从ROCC协处理器端传递进来的指令数据；通过判断指令数据的类型序号，将指令数据分类；并将指令数据相关信息存放在对应的指令存储单元存放到指令存储单元；指令数据相关信息包括每组指令数据的地址和数据的长度；

[0010] B. 输入数据存储器用于在接口行为状态机和读入数据状态机的协同作用下，与L1 cache端建立握手协议，接收L1 cache端发来的数据并保存；当输入数据存储器中的数据存满后，发送给ROCC协处理器端，可对输入数据存储器的位宽和深度进行调整；

[0011] C. 计算结果存储器用于在接口行为状态机作用下，接收ROCC协处理器传递来的计算结果数据并保存；当全部传递完成后，通过与L1 cache端建立握手协议，将结果数据发送给L1 cache端；

[0012] D. 读入数据状态机用于调度输入数据从L1 cache端到输入数据存储器，可通过对指令解析模块得到的数据寄存器数组（指令存储单元）的遍历访问，切换L1 cache端的数据地址，并根据每组数据长度信息，实现多组数据的传输；

[0013] E. 接口行为状态机用于通过定义多个状态实现协处理器的数据搬运和协处理器驱动。

[0014] 针对上述的ROCC协处理器接口模型，其中的：

[0015] 指令解析模块中，通过判断从ROCC协处理器端传递进来的指令数据的类型序号，将指令数据分类；类别包括：传递数据的组数、配置协处理器数据、计算数据、结果返回地址、直接传递指令。

[0016] 计算结果存储器的位宽和深度可根据协处理器计算结果的大小进行调整。

[0017] 读入数据状态机包括多个状态：m_idle、m_read、m_pad和m_absorb，分别表示：

[0018] m_idle为初始空闲状态；

[0019] m_read：当输入数据的位宽与输入数据存储器的位宽相同时，从L1 cache端读入到输入数据存储器；

[0020] m_pad：当输入数据的位宽小于输入数据存储器的位宽时，从L1 cache端读入输入数据存储器，通过比较输入数据位宽与buffer_in输入数据存储器位宽的差异，进行空缺位补零，并添加数据结束标志位操作；

[0021] m_absorb只用于m_pad的下一个状态；与m_pad共同用于传递小于buffer_in位宽数据即一组数据中末端的数据。

[0022] 接口行为状态机包括多个状态：s_idle、s_absorb、s_finish、s_calculate、s_store和s_write；其中：

[0023] s_absorb状态：在读入数据状态机m_state的作用下调度读入数据，当buffer_in存储L1 cache端数据存满时，接口行为状态机s_state的状态由s_idle切换到s_absorb，将buffer_in中的数据传递到协处理器，将buffer_in中数据全部传输完成后进入到s_finish

状态进行判断,判断是否所有配置数据是否传递完成,若完成则进入s_calculate状态,若未完成则继续返回到s_idle状态,等待buffer_in数据存满,进行下一轮向下处理器传递数据;

[0024] 在经过判断配置数据全部传递完成,即由s_finish状态进入到s_calculate状态;

[0025] 在s_calculate状态时,包括:a.将输入数据从L1 cache端传输到buffer_in,再从buffer_in传输到ROCC协处理器;b.通过协处理器对不断接收的计算数据进行运算操作,直至所有计算数据传递完成;所有计算数据被运算完成后,由协处理器发出完成运算信号,由s_calculate状态跳转到运算结果数据存储状态s_store;

[0026] 在存储状态s_store时,通过接口模型与ROCC协处理器端建立起握手信号,开始进行运算结果数据返回,从协处理器端传递到计算结果存储器buffer_out;

[0027] 在完成数据存储后,进入到数据写回状态s_write,通过接口与L1 cache端建立握手,运算结果数据由buffer_out传递到指令存储单元保存的输出结果地址处。

[0028] 本发明还提供了一种ROCC协处理器接口模型的自动生成工具,包括:接口生成功能实现inst_gen.hpp和接口生成主函数main.cpp;接口生成主函数main.cpp用于接收用户对接口模型的相关配置参数;接口生成功能实现inst_gen.hpp用于定义接口生成功能实现的多个函数;通过调用接口生成功能实现inst_gen.hpp中的函数,实现的ROCC协处理器接口模型的生成。

[0029] 上述的ROCC协处理器接口模型的自动生成工具的实现方法包括如下步骤:

[0030] 1)用户自定义协处理器;

[0031] 确定用户自定义协处理器的输入数据和输出数据信息;其中,用户自定义数据包括输入数据指令数data_num、输入数据地址data_addr、输入数据长度data_len、输出结果地址output_addr、直接传递到协处理器端的指令数值direct_value、输入数据存储器buffer_in和计算结果存储器buffer_out大小数值;

[0032] 2)根据用户自定义协处理器的输入数据,确定指令解析的次数和指令存储单元的大小,并定义输入数据存储器buffer_in和计算结果存储器buffer_out的大小;

[0033] 3)设计ROCC协处理器接口模型生成的主函数main.cpp,通过判断用户输入数据指令中各个字段位置的不同,对用户输入数据指令采用多种方法进行指令解析,获取一组数据的数据地址和数据长度;

[0034] 4)根据用户自定义的输入数据,确定从内存L1 cache端传送数据进入接口模型的数据组数;再根据指令数据的长度信息,生成不同次数的切换状态;

[0035] 通过用户给定的输入数据组数data_num值,确定生成的接口模型要从内存L1 cache端传送多少组数据进入接口模型;

[0036] 根据指令数据的长度信息,进行读入数据的地址和长度信息的切换,根据用户给定的输入数据组数data_num值的不同,在接口模型中依次对data_num组数据进行传送;

[0037] 5)通过用户给定的输入数据组数num值,定义传送行为的状态,生成接口模型的行为状态机;

[0038] 将用户自定义的输入数据中的数据组数data_num值作为s_finish状态的状态跳转判断条件,若现在传递数据组数数值刚好为data_num,进入s_calculate状态;其余状态为固定形式,描述采用字符输出;行为状态机的状态包括:s_idle、s_absorb、s_finish、s_

calculate、s_store和s_write;其中在s_finish状态时进行判断,判断是否所有配置数据是否传递完成,若完成则进入s_calculate状态,若未完成则继续返回到s_idle状态;

[0039] 通过上述步骤2)~5),即可生成ROCC协处理器接口模型文件,实现对加速器开发人员简化和屏蔽ROCC协处理器接口模型;

[0040] 6)通过对用户自定义的输入数据中的指令名和指令给定的顺序和个数进行识别,将用户给定的简化指令转换成RISC-V自定义ISA,生成指令描述文件;

[0041] 通过上述步骤,可得到协处理器接口文件和指令描述头文件,即构建得到ROCC协处理器接口模型的自动生成工具;通过利用ROCC协处理器接口模型的自动生成工具,实现ROCC协处理器接口模型的自动生成。

[0042] 针对上述的ROCC协处理器接口模型的自动生成工具的实现方法,步骤3)中,通过判断用户输入数据指令中各个字段位置的不同,对用户输入数据指令采用多种方法进行指令解析,包括如下解析过程:

[0043] a.对于从ROCC协处理器端获取信息的指令im.create_instr("DATA", "data1_addr", "data1_len"),指令中"DATA", "data1_addr", "data1_len"分别为第一字段、第二字段和第三字段,第一字段为指令名;第二字段和第三字段位置均非空;解析该指令时,从ROCC协处理器端获取rs1和rs2两个值;rs1,rs2为RISC-V自定义ISA中的源寄存器;

[0044] b.对于从ROCC协处理器端获取信息的指令im.create_instr("OUTPUT", "output_addr"),指令中"OUTPUT", "output_addr"分别为第一字段、第二字段;第一字段为指令名;第二字段位置非空;解析该指令时,从ROCC协处理器端获取rs1,输出rs1值。

[0045] 针对上述的ROCC协处理器接口模型的自动生成工具的实现方法,用户自定义的输入数据中的指令名包括DATA、OUTPUT或DIRECT_VALUE;可根据用户给出指令的先后次序识别指令的顺序和个数,将用户给定的简化指令转换成RISC-V自定义ISA。

[0046] 针对上述的ROCC协处理器接口模型的自动生成工具的实现方法,步骤6)中,将用户给定的简化指令转换成RISC-V自定义ISA的格式包括:costomx rd,rs1,rs2,funct;具体实施时,包括如下实施过程:

[0047] 用户给定:im.create_instr("DATA", "data1_addr", "data1_len");

[0048] 生成ISA:#define DATA(arg1, arg2) \

[0049] ROCC_INSTRUCTION_DSS(2, 0, arg1, arg2, DATA_ID)

[0050] 即生成指令描述文件。

[0051] 与现有技术相比,本发明的有益效果包括:

[0052] 利用本发明提供的技术方案,可以快速为可定制协处理器提供接口模型,降低用户将自定义加速器集成到RISC-V系统框架的时间开销,提高了开发人员测试效率。

附图说明

[0053] 图1为本发明的ROCC协处理器接口模型设计结构框图。

[0054] 图2为本发明的ROCC协处理器接口模型的读入数据状态机的结构框图。

[0055] 图3为本发明的ROCC协处理器接口模型的行为状态机的结构框图。

[0056] 图4为本发明的自动生成接口模型的流程框图。

[0057] 图5为本发明的ROCC协处理器集成到RISC-V系统运行流程框图。

[0058] 图6为本发明的ROCC协处理器集成到RISC-V系统的结构框图。

具体实施方式

[0059] 下面结合附图,通过实施例进一步描述本发明,但不以任何方式限制本发明的范围。

[0060] 本发明提供一种ROCC协处理器接口模型,如图1所示,包括:指令解析模块、指令存储单元、输入数据存储器buffer_in、计算结果存储器buffer_out、读入数据状态机模块m_state和接口行为状态机模块s_state。

[0061] 1. 指令解析模块与ROCC端口相连接;指令解析模块接收从ROCC端传递进来的指令数据,判断是否满足指令解析的条件,若满足条件判断传递的指令类型序号,将用户自定义的输入数据中的指令数据(包括DATA、OUTPUT或DIRECT_VALUE)分类:传递给协处理器的配置或计算数据、结果返回地址、直接传递指令等,并将每组数据地址、数据长度存放在加速器对应的指令存储单元里,待后续处理使用。

[0062] 2. 输入数据存储器buffer_in与ROCC协处理器接口模型之外的内存L1 cache相连,用于在接口行为状态机s_state和读入数据状态机m_state的协同作用下,与L1 cache端建立握手协议,接收L1 cache端发来的数据并保存,当buffer_in数据存满后,发送给ROCC协处理器(图1中的加速器acc),用户可以根据协处理器的需求,对buffer_in的位宽和深度进行调整。

[0063] 3. 计算结果存储器buffer_out与ROCC协处理器接口外的内存L1 cache相连,用于在接口行为状态机s_state作用下,当ROCC协处理器数据处理完成,接收协处理器发来的计算结果数据并保存,全部传递完成后,通过与L1 cache端建立握手协议,将结果数据发送给L1 cache端。buffer_out的位宽和深度同样可以根据协处理器计算结果的大小调整。

[0064] 4. 读入数据状态机m_state(如图2),主要负责输入数据从L1 cache端到buffer_in的调度,读入数据状态机m_state分为4个状态:m_idle、m_read、m_pad和m_absorb,其中m_read用于输入数据与buffer_in位宽相同时,从L1 cache端读入到输入数据存储器buffer_in;m_pad用于输入数据位宽小于buffer_in位宽时,从L1 cache端读入buffer_in,通过比较输入数据位宽与buffer_in位宽差异,进行空缺位补零,并添加数据结束标志位操作;m_absorb专用于m_pad的下一个状态,与m_pad共同作用,用于传递小于buffer_in位宽数据即一组数据中末端的数据。读入数据状态机m_state可通过对1中指令解析模块得到的数据寄存器数组(指令存储单元)的遍历访问,切换L1 cache端数据地址,并根据每组数据长度信息,实现多组数据传输过程。

[0065] 5. 接口行为状态机s_state(如图3),是ROCC协处理器接口模型的核心模块,主要负责所有数据搬运、协处理器驱动等问题。接口行为状态机s_state主要分为6个状态:s_idle、s_absorb、s_finish、s_calculate、s_store和s_write,其中s_absorb状态在读入数据状态机m_state的作用下调度读入数据,当buffer_in存储L1 cache端数据存满时,接口行为状态机s_state的状态由s_idle切换到s_absorb,将buffer_in中的数据传递到协处理器,将buffer_in中数据全部传输完成后进入到s_finish状态进行判断,判断是否所有配置数据是否传递完成,若完成则进入s_calculate状态,若未完成则继续返回到s_idle状态,等待buffer_in数据存满,进行下一轮向协处理器传递数据;在经过判断配置数据全部传递

完成,即可由s_finish状态进入到s_calculate状态,在这个状态时,一边是输入数据从L1 cache端传输到buffer_in,再从buffer_in传输到ROCC协处理器,一边是协处理器对不断接收的计算数据进行运算操作,直至所有计算数据传递完成,所有计算数据被运算完成,等待由协处理器发出的完成运算信号,接收到这个完成信号后,即可由s_calculate状态跳转到运算结果数据存储状态s_store;在存储状态s_store时,通过接口模型与ROCC协处理器建立起来的握手信号,开始进行运算结果数据返回,从协处理器端传递到计算结果存储buffer_out,由此,要求用户根据协处理器端运算结果数据的大小,对buffer_out的位宽和深度进行设定;在完成数据存储后,进入到s_write数据写回状态,通过接口与L1 cache端建立握手,运算结果数据由buffer_out传递到l中指令解析模块保存的输出结果地址处。

[0066] 如图1所示的框图中包括4个组成部分:其中,ROCC是Chipyard开源框架的RISC-V内核的RoCC端;Memory是内存系统即L1高速缓存;ctrl是本发明设计的ROCC协处理器接口模型;acc是用户自定义的协处理器。本发明设计的ROCC协处理器接口模型与RISC-V内核的RoCC端之间的接口连接信息采用图1中的①ROCC Cmd,包括了握手信号rocc_req_val和rocc_req_rdy、源寄存器rocc_rs1和rocc_rs2、目的寄存器rocc_rd、功能位rocc_funct;本发明设计的ROCC协处理器接口模型与Memory(是内存系统)之间的连接包括从Memory向输入数据存储器buffer_in传输数据的数据通路dmem_data、从计算结果存储器buffer_out向Memory传输结果数据的数据通路res_toMem和图1中的②握手、计数和地址信息等,包括:从Memory向输入数据存储器buffer_in传输数据的握手信号dmem_req_val和dmem_req_rdy、计数信号dmem_req_tag、地址dmem_req_addr、命令dmem_req_cmd、数据大小dmem_req_size,从计算结果存储器buffer_out向Memory传输结果数据握手dmem_resp_val、计数dmem_resp_tag、数据dmem_resp_dat。本发明设计的ROCC协处理器接口模型ctrl与用户自定义的协处理器acc之间的接口信息包括:从ROCC协处理器接口模型ctrl中的指令存储单元中到直接传递到用户自定义的协处理器acc的immediate,传递协处理器端的指令数值direct_value;从ROCC协处理器接口模型的行为状态机与用户自定义的协处理器acc的表示当前正在传递的数据组数信号num、加速器启动信号acc_start,ROCC协处理器接口模型与用户自定义的协处理器acc之间的传送数据和返回结果数据的握手信号,包括:ROCC协处理器接口模型向用户自定义的协处理器acc传送数据握手buffer_outVal和buffer_outRdy,用户自定义的协处理器acc向ROCC协处理器接口模型返回数据握手res_returnVal和res_returnRdy;两组数据传输通路及其计数信号:ROCC协处理器接口模型向用户自定义的协处理器acc传送数据通路信号buffer_out和计数信号aindex,用户自定义的协处理器acc向ROCC协处理器接口模型返回数据通路信号res_return和计数信号windex。

[0067] 具体实施时,本发明实现了一种ROCC协处理器接口模型自动生成工具,整个ROCC协处理器接口模型自动生成工具包括两个部分:接口生成功能实现inst_gen.hpp和接口生成的主函数main.cpp。用户将接口模型的相关配置参数在主函数中给定,通过调用接口生成功能实现的一些函数,实现接口的生成;ROCC协处理器接口模型自动生成工具的实现方法包括如下步骤:

[0068] 1) 确定用户自定义协处理器的输入数据和输出数据信息;

[0069] 用户自定义数据包括输入数据指令数data_num、输入数据地址data_addr、输入数据长度data_len、输出结果地址output_addr、直接传递到协处理器端的指令数值direct_

value、输入数据存储器buffer_in和计算结果存储器buffer_out大小数值,在接口生成的主函数main.cpp中给定;

[0070] 2)根据用户自定义的输入数据,确定指令解析的次数和指令存储单元的大小,并定义输入数据存储器buffer_in和计算结果存储器buffer_out大小;

[0071] 通过在接口生成的主函数main.cpp中给出上面提到的用户自定义接口信息,接口生成功能实现inst_gen.hpp获取输入数据指令数data_num,由此确定接口模型指令解析模块的解析次数和指令存储单元的大小,接口生成功能实现inst_gen.hpp获取输入数据存储器buffer_in和计算结果存储器buffer_out大小数值,接口模型定义输入数据存储器buffer_in和计算结果存储器buffer_out;

[0072] 3)对用户输入的多条数据指令进行解析;

[0073] 通过判断在接口生成的主函数main.cpp中给出的指令中各个字段位置的不同,进行不同方式的指令解析,判断每条指令有几个字段,一般有两个或三个字段,第一个字段为指令名,第二和第三个字段是ROCC端传递的信息;若有两个字段,则从ROCC端获取rs1寄存器中的信息,存放在指令存储单元;若有三个字段,则从ROCC端获取rs1和rs2两个寄存器中的信息,存放在指令存储单元。例如:

[0074] 指令im.create_instr("DATA", "data1_addr", "data1_len");

[0075] 指令中"DATA", "data1_addr", "data1_len"分别为第一字段、第二字段和第三字段,第一字段为指令名,经过判断第二字段和第三字段位置都非空,因此这条指令要从从ROCC端获取两条有用信息,那么解析这条指令时,需要从ROCC端获取rs1和rs2(RISC-V自定义ISA中rs1,rs2 为源寄存器)两个值,存放在指令存储单元。通过这类指令解析,可以实现DATA类型的指令解析,同时获取一组数据的数据地址data_addr和数据长度data_len;

[0076] 另一种,如指令im.create_instr("OUTPUT", "output_addr");

[0077] 指令中"OUTPUT", "output_addr"分别为第一字段、第二字段,第一字段为指令名,经过判断第二字段位置都非空,因此这条指令要从从ROCC端获取一条有用信息,那么解析这条指令时,需要从ROCC端获取rs1输出一个值,存放在指令存储单元,通过这类指令解析,可以实现OUTPUT类型的指令解析结果,获取输出地址output_addr,同理适用于DIRECT_VALUE指令;

[0078] 4)根据用户自定义的输入数据,确定从内存L1 cache端传送数据进入接口模型的数据组数;再根据指令存储单元中存储的数据长度信息,生成不同次数的切换状态;

[0079] 通过用户给定的输入数据组数data_num值,自动生成工具可以确定生成的接口模型要从内存L1 cache端传送多少组数据进入接口,由于每组数据的地址和长度又是不同的,因此在接口模型的读入数据状态机m_state里面根据指令存储单元中存储的数据长度信息,进行读入数据的地址和长度信息的切换,根据用户给定的输入数据组数data_num值的不同,在接口模型中依次对data_num组数据传送;

[0080] 5)行为状态机主要有6个状态:s_idle、s_absorb、s_finish、s_calculate、s_store和s_write,其中在s_finish状态进行判断,判断是否所有配置数据是否传递完成,若完成则进入s_calculate状态,若未完成则继续返回到s_idle状态。这里将用户自定义的输入数据中的数据组数data_num值作为s_finish状态的状态跳转判断条件,若现在传递数据组数数值刚好为data_num,进入s_calculate状态。其余状态为固定形式,描述采用字符输

出。组合起来生成行为状态机；

[0081] 通过用户给定的输入数据组数num值,自动生成工具生成的接口模型的行为状态机,同样进行与num数值相关的状态跳转,过程可参考实施例1;

[0082] 6)通过对用户自定义的输入数据中的指令名(DATA、OUTPUT或DIRECT_VALUE)和指令给定的顺序和个数(ID号)进行识别(可根据用户在在接口生成的主函数main.cpp中给出的指令的先后次序识别指令给定的顺序和个数(ID号)),将用户给定的简化指令转换成RISC-V自定义ISA;

[0083] 通过上述步骤2)、3)、4)、5),即可生成ROCC协处理器接口模型文件,实现了对加速器开发人员简化和屏蔽ROCC协处理器接口的具体实现细节,另外,还需对加速器开发人员简化和屏蔽RISC-V指令。自动生成工具通过识别简化指令名(DATA、OUTPUT或DIRECT_VALUE)(指令的第一字段内容)和指令给定的顺序和个数(ID号),将用户给定的简化指令转换成RISC-V自定义ISA(格式costomx rd,rs1,rs2,funct),例如:

[0084] 用户给定:im.create_instr("DATA", "data1_addr", "data1_len");

[0085] 生成ISA:#define DATA(arg1, arg2) \

[0086] ROCC_INSTRUCTION_DSS(2, 0, arg1, arg2, DATA_ID)

[0087] 生成指令描述文件。

[0088] 通过上述步骤,可得到协处理器接口文件(ctrl.scala)和指令描述头文件(instr.h),实现ROCC协处理器接口模型的自动生成,即构建得到ROCC协处理器接口模型自动生成工具。

[0089] 具体实施时,利用上述ROCC协处理器接口模型自动生成工具生成ROCC协处理器接口模型,首先生成ROCC协处理器接口文件和指令头文件,再将ROCC协处理器集成到RISC-V系统;具体包括如下步骤:

[0090] 1.在生成协处理器接口文件和指令头文件时,执行如下操作:

[0091] 1.1确定用户自定义协处理器的输入数据和输出数据信息,包括输入数据指令数(data_num)、输入数据地址(data1_addr、data2_addr、等等)、输入数据长度(data1_len、data2_len、等等)、输出结果地址(output_addr)、直接传递到协处理器端的指令数值(direct_value)等;

[0092] 1.2 确定ROCC协处理器接口模型的输入数据存储器buffer_in和计算结果存储器buffer_out大小;

[0093] 1.3 在接口模型自动生成工具中给出用户信息的命令描述,主要是在自动生成工具代码主函数中描述1.1的信息(参实施例1);

[0094] 1.4 运行自动生成工具程序,得到协处理器接口文件(ctrl.scala)和指令描述头文件(instr.h)。

[0095] 2.把用户自定义的ROCC协处理器集成到RISC-V系统时,执行如下操作:

[0096] 2.1将自动生成工具生成的协处理器接口文件(ctrl.scala)和指令描述头文件(instr.h)分别放在硬件源代码位置和测试的应用程序文件夹处,指令描述头文件(instr.h)作为应用程序文件的头文件(如图6);

[0097] 2.2在应用程序文件中给出简明的用户自定义协处理器的输入数据地址和长度信息、输出数据地址以及数据组数信息,如:对应自动生成工具代码主函数中描述im.create_

instr("DATA", "data1_addr", "data1_len"),这里主要调用指令描述文件(instr.h),用简明语言描述指令DATA(&data1, sizeof(data1))(参实施例1);

[0098] 2.3编译应用程序文件,生成不调用自定义ISA的汇编.riscv文件(参实施例1);

[0099] 2.4构建该协处理器的Chipyard配置,用Verilator模拟仿真。

[0100] 下面通过实例对本发明做进一步说明。

[0101] 实施例1:

[0102] 利用本发明实现的ROCC协处理器接口模型自动生成工具,将功能为矩阵乘加运算的加速器(作为ROCC协处理器)集成到RISC-V系统框架,包括如下步骤:

[0103] 在生成协处理器接口文件和指令头文件时,执行如下操作:

[0104] 1) 确认实现矩阵乘加运算(mA+B)的输入数据指令数一共有2条,即num=2。其中包括:矩阵A的数据地址和长度(data1_addr、data1_len),矩阵B的数据地址和长度(data2_addr、data2_len)。另外还有矩阵乘的系数m的数值(direct_value)和输出结果的地址(output_addr);

[0105] 2) 根据用户自定义协处理器的要处理的数据大小,以及处理后结果数据的大小,确认接口模型输入数据存储器buffer_in和计算结果存储器buffer_out大小,其中由于传递数据时,输入数据存储器buffer_in可以进行多轮传递,因此buffer_in的大小设定可以小于一组数据的大小,尽量不要大于一组数据的大小,造成面积浪费;计算结果存储器buffer_out大小要大于等于计算结果数据的大小,保证结果数据完整正确存储;

[0106] 3) 在接口模型自动生成工具中给出用户信息的命令描述,主要是在自动生成工具代码主函数中描述1)的信息,例如:在接口生成的主函数main.cpp中,先声明一个指令管理的类 InstrManager im,然后调用不同的方法,分别给出输入数据指令数(data_num)、输入数据地址(data1_addr、data2_addr)、输入数据长度(data1_len、data2_len)、输出结果地址(output_addr)、直接传递到协处理器端的指令数值(direct_value)等,最后调用finalize方法;

[0107] 示意代码如下:

```
[0108] int main() {
[0109]     InstrManager im;
[0110]     im.add_data_num(2);
[0111]     im.buffer_in_depth(16);
[0112]     im.buffer_out_depth(16);
[0113]     im.create_instr("DATA", "data1_addr", "data1_len");
[0114]     im.create_instr("DATA", "data2_addr", "data2_len");
[0115]     im.create_instr("OUTPUT", "output_addr");
[0116]     im.create_instr("DIRECT_VALUE", "m");
[0117]     im.finalize();
[0118]     return 0;
[0119] }
```

[0120] 4) 运行自动生成工具程序,得到协处理器接口文件(ctrl.scala)和指令描述头文件(instr.h)。

[0121] 把ROCC协处理器集成到RISC-V系统时,执行如下操作:

[0122] 1) 将自动生成工具生成的接口文件ctrl.scala放在硬件源代码位置(~src/main/scala),指令描述头文件instr.h放在测试的应用程序文件夹(~software/tests/src)下,作为应用程序文件的头文件;

[0123] 2) 在应用程序文件中,通过调用头文件instr.h的宏定义指令,给出简明的用户自定义协处理器的输入数据地址和长度信息、输出数据地址以及数据组数信息;

[0124] 示意代码如下:

```
[0125] unsigned char data1[128] = {'\1' , '\2' , '\3' , '\4' , ...};
```

```
[0126] unsigned char data2[128] = {'\1' , '\2' , '\3' , '\4' , ...};
```

```
[0127] unsigned char m = {'\3' };
```

```
[0128] unsigned char output[128] ;
```

```
[0129] NUM(2);
```

```
[0130] DATA(&data1, sizeof(data1));
```

```
[0131] DATA(&data2, sizeof(data2));
```

```
[0132] OUTPUT(&output);
```

```
[0133] DIRECT_VALUE(m);
```

```
[0134] EMPTY();
```

[0135] 3) 编译应用程序文件生成自定义ISA的汇编.riscv文件,进入到~chipyard/generators/muladd/software路径下,执行./build,可以看到在~software/tests/bare路径下生成了我们所需要的.riscv文件;

[0136] 4) 构建该协处理器的Chipyard配置,用Verilator模拟仿真,进入到~chipyard/sims/verilator路径下,执行命令:make CONFIG=MuladdRocketConfig run-binary BINARY=/root/chipyard/generators/muladd/software/tests/bare/muladd-rocc.riscv,其中make生成可执行文件,run-binary执行软件编译生成的二进制文件。

[0137] 通过以上步骤,用户就可以把自定义加速器集成到RISC-V系统框架进行测试了。

[0138] 需要注意的是,公布实施例的目的在于帮助进一步理解本发明,但是本领域的技术人员可以理解:在不脱离本发明及所附权利要求的范围内,各种替换和修改都是可能的。因此,本发明不应局限于实施例所公开的内容,本发明要求保护的范围以权利要求书界定的范围为准。

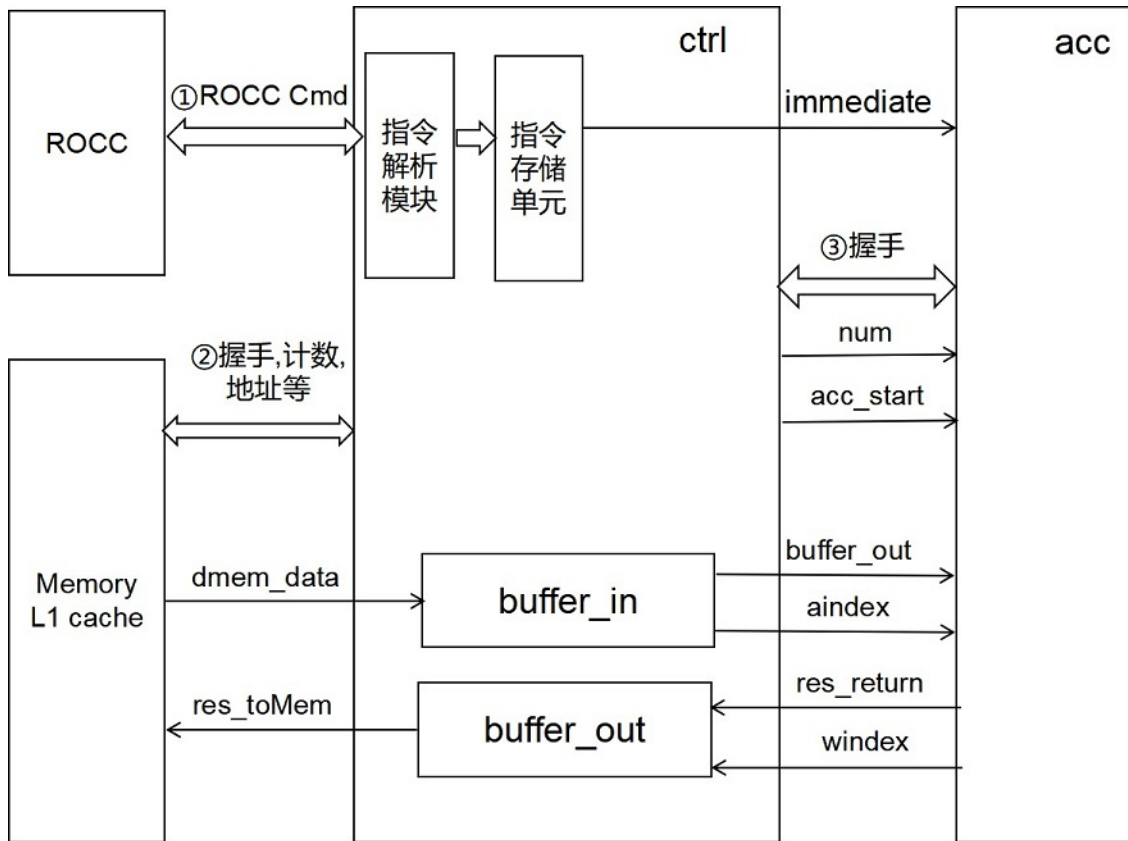


图1

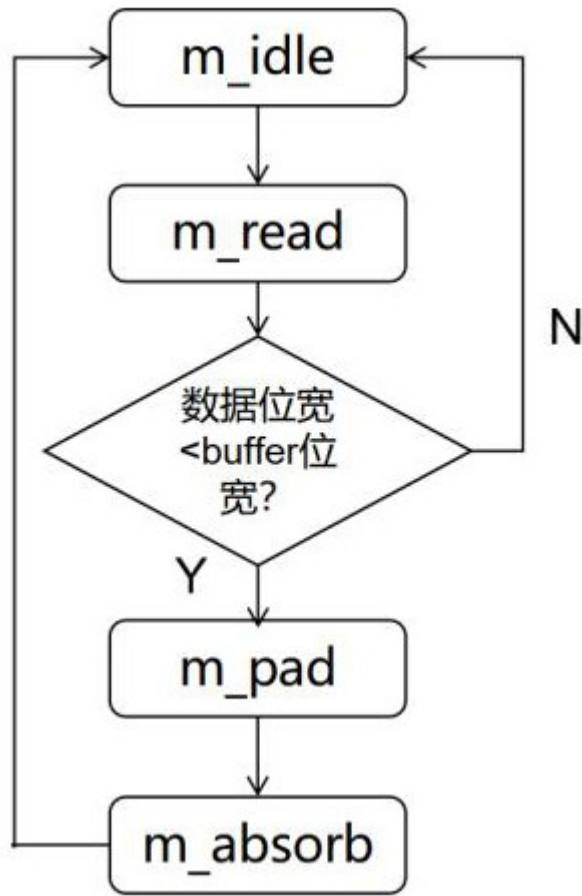


图2

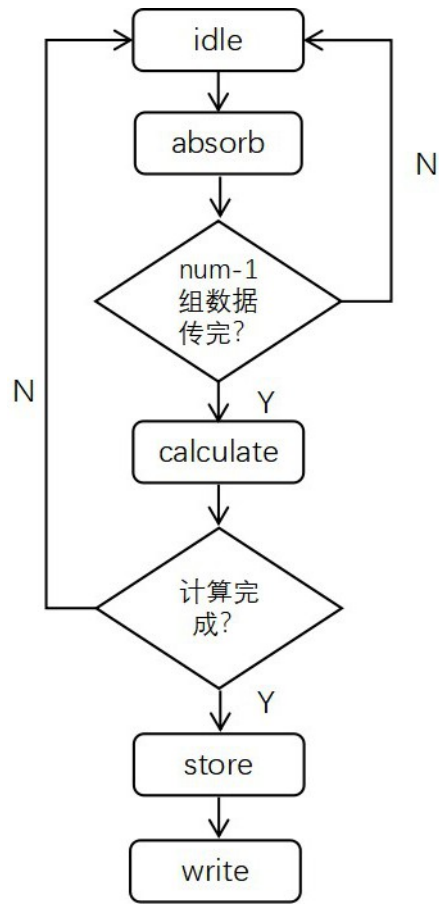


图3

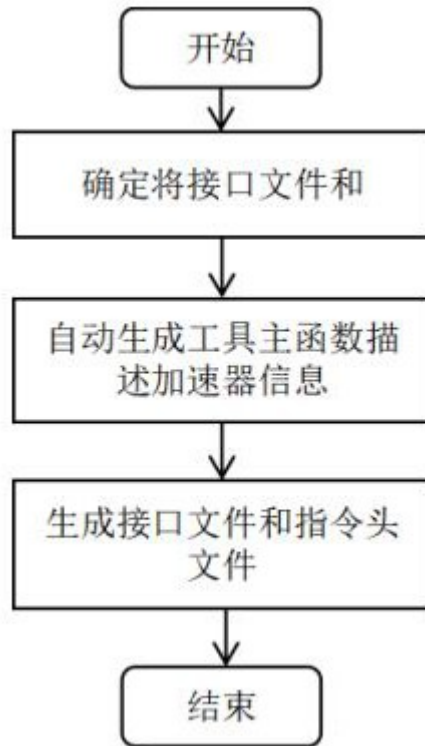


图4

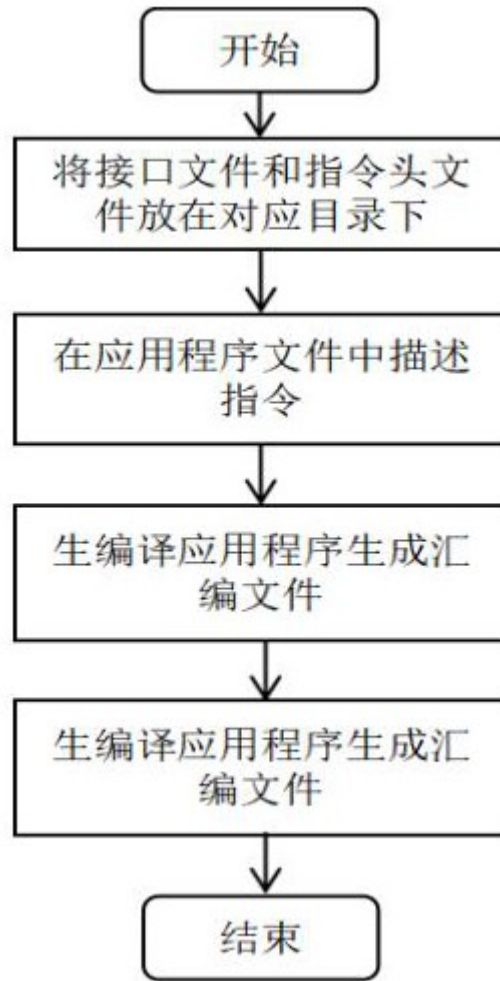


图5

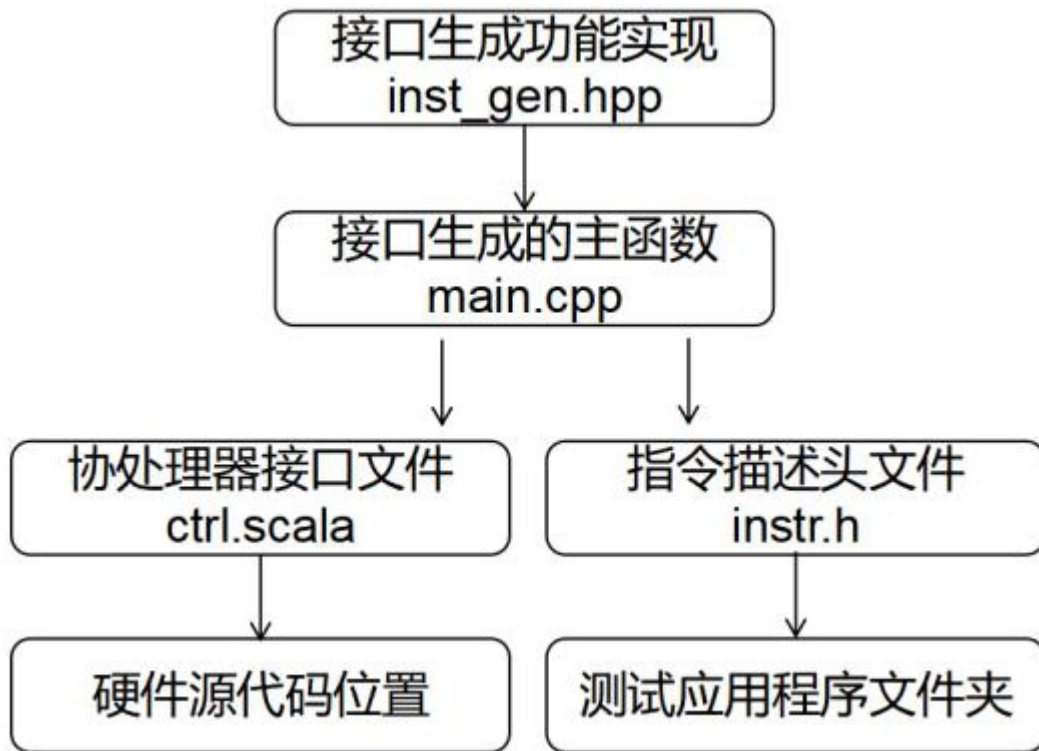


图6