|  |  |
| --- | --- |
| 文档编号 |  |
| 文档版本 |  |
| 文档管控 | 内部公开 |
| 存档日期 |  |

昆明湖V1项目IBuffer模块AS

|  |  |
| --- | --- |
| 编 写： | 满洋 |
| 校 对： |  |
| 审 核： |  |
| 批 准： |  |

昆明湖V1项目

2024年01月01日

文档修订记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **版本编号** | **变化状态** | **变更说明** | **作者** | **日期** |
| 1 | V0.1 | C |  | 满洋 | 2024-01-01 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

\*变化状态：C—创建，A—增加，M—修改，D—删除

文档审批信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **审核** | **会签** | **批准** | **备注** |
| V1.0 |  |  |  |  |
|  |  |  |  |  |

目 录

[1 简介 1](#_Toc155629402)

[1.1 文档介绍 1](#_Toc155629403)

[1.2 参考文档 1](#_Toc155629404)

[1.3 术语说明 1](#_Toc155629405)

[2 设计规格 1](#_Toc155629406)

[3 功能描述 1](#_Toc155629407)

[3.1 功能概述 1](#_Toc155629408)

[3.2 功能详述 2](#_Toc155629409)

[3.2.1 接收IFU送入的指令 2](#_Toc155629410)

[3.2.2 提供给后端Decode模块指令 2](#_Toc155629411)

[4 总体设计 2](#_Toc155629412)

[4.1 整体框图 2](#_Toc155629413)

[4.2 接口列表 2](#_Toc155629414)

[4.3 接口时序 3](#_Toc155629415)

[4.4 时钟复位 3](#_Toc155629416)

[4.5 寄存器配置 3](#_Toc155629417)

[5 模块设计 3](#_Toc155629418)

[6 PPA 4](#_Toc155629419)

[6.1 时序 4](#_Toc155629420)

[6.2 面积 4](#_Toc155629421)

[7 验证关注点 4](#_Toc155629422)

[8 Floorplan 建议 4](#_Toc155629423)

[9 遗留问题 4](#_Toc155629424)

1. 简介
   1. 文档介绍

本文档是昆明湖V1项目2023年11月RTL Freeze的AS文档，描述前端取指模块中IBuffer子模块的架构设计。

* 1. 参考文档

NA

* 1. 术语说明

表1.1 术语说明

|  |  |  |
| --- | --- | --- |
| **缩写** | **全称** | **描述** |
| IBuffer | Instruction Buffer | 指令缓冲 |
| PC | Program Counter | 程序计数器 |
| MUX | Multiplexer | 数据选择器/多路复用器 |
| - | FetchWidth | 取指宽度（处理器设计参数） |
| - | DecodeWidth | 译码宽度（处理器设计参数） |
| IBufNBank | Instruction Buffer Bank Amount | 指令缓冲分Bank数量（处理器设计参数） |
| IBufSize | Instruction Buffer Size | 指令缓冲大小（处理器设计参数） |

1. 设计规格
2. 支持从IFU模块接收指令
3. 支持向后端送出指令
4. 功能描述
   1. 功能概述

IBuffer模块接收IFU模块送入的指令，并存入内部的队列。指令出队后进入后端Decode模块。

当收到重定向冲刷信号时，恢复指针至复位值。

* 1. 功能详述
     1. 接收IFU送入的指令

IBuffer模块可接收从IFU送入的指令，每周期最多16条（FetchWidth）（如全部为C扩展压缩指令）。

* + 1. 提供给后端Decode模块指令

IBuffer模块向后端Decode模块提供指令，每周期最多6条（DecodeWidth）。

1. 总体设计

总体分为三个重要部分，队列写口，队列读口和指针维护。

* 1. 整体框图

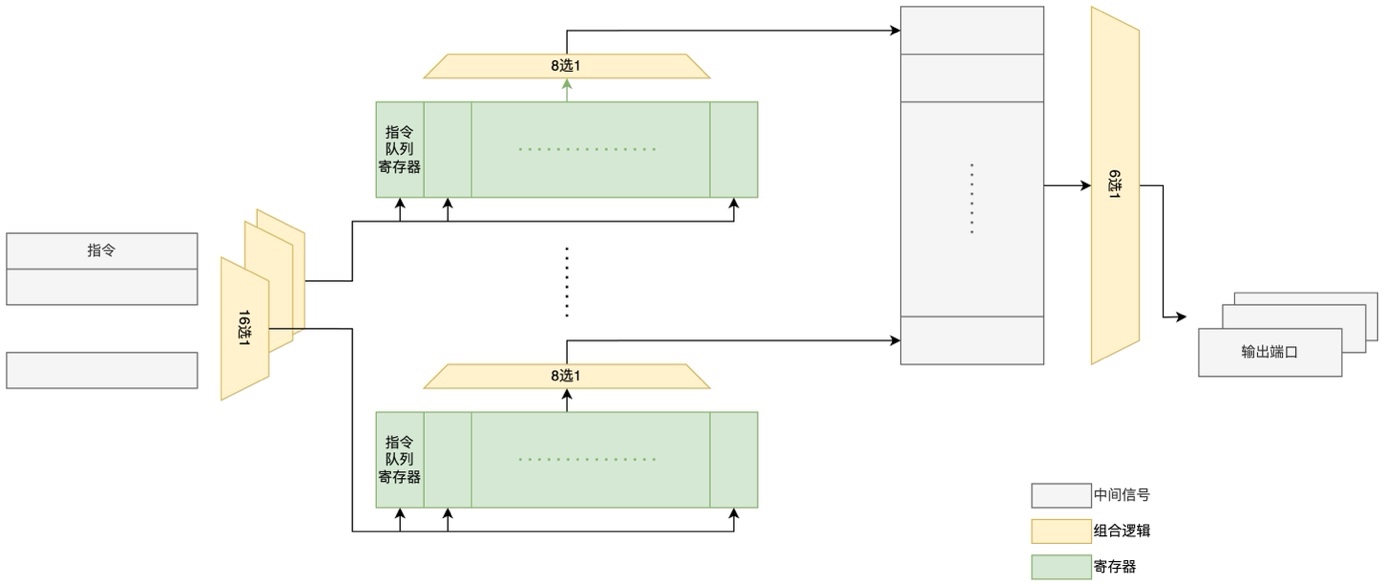
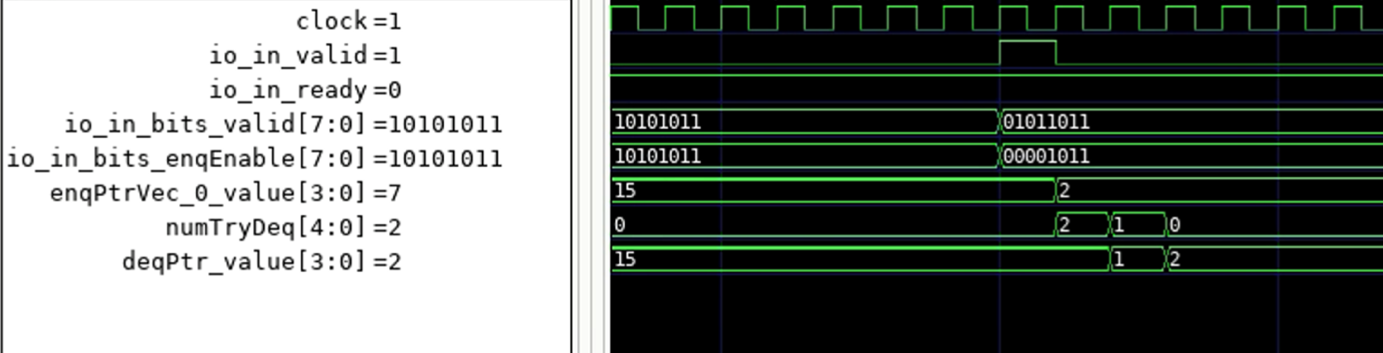


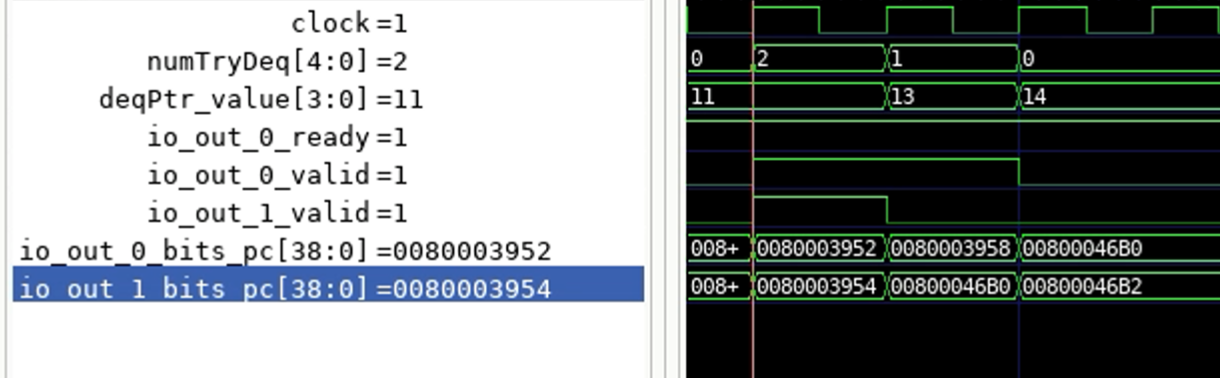
图 1

* 1. 接口列表

见IBuffer接口文档。

* 1. 接口时序





* 1. 时钟复位

|  |  |  |
| --- | --- | --- |
| Module | Clock | Reset |
| IBuffer | clock | reset |

* 1. 寄存器配置

NA

1. 模块设计

队列大小为IBufSize，分为IBufNBank个Bank，使用寄存器实现。对于昆明湖，寄存器堆组织形式是8 Entry \* 6 Bank。

1. 队列写口：因为每条从IFU进入的指令都有可能写入每一个Entry，因此实现形式是每个Entry使用一个MUX，从IFU进入的所有指令中，根据指针和IFU传入的valid、enqOffset等信号控制写使能和选择数据。
2. 队列读口：因为从IBuffer出队的指令必然是连续的，因此可以利用这个特点节省组合逻辑。读口根据Bank内指针，首先从Bank内选择1条指令，再从所有Bank中根据Bank间指针对指令进行Shuffle，最后送到端口。
3. 指针维护
   1. 队列间指针维护：根据后端的ready信号，每周期加6（DecodeWidth）或0
   2. 队列内指针维护：根据本周期选出的指令是否同时有效和被接受（fire），决定加1或0
4. PPA
   1. 时序

为减少指针扇出（fanout），每个读端口的MUX使用一个寄存器作为指针。同样，每个写口的MUX也使用一个寄存器作为指针。复位时，这些指针复位成0~N-1（对应端口数量）。

* 1. 面积

读口使用了分Bank设计，相比直接读整个队列的做法，组合逻辑由6个48选1减少至6个8选1+6个6选1，能够节省面积。

1. 验证关注点
2. 队列的指针维护正确性
3. 后端接收指令的设计假设，目前IBuffer设计假设后端ready信号给出的当拍，所有送到端口上的指令都被接收。
4. Floorplan 建议

模块除了接收重定向信号，仅与IFU和后端Decode模块交互，因此建议放置在Frontend和Backend之间，一般情况只需要由EDA工具确定即可，不需要特殊处理。

1. 遗留问题

NA