**高能效的ECG检测CNN加速器验证报告**

**目录**

[修改记录 3](#_Toc56430710)

[1. 文档信息 4](#_Toc56430711)

[1.1 功能模块信息 4](#_Toc56430712)

[1.2 版本 4](#_Toc56430713)

[1.3 参考文档 4](#_Toc56430714)

[2. 模块设计 5](#_Toc56430715)

[2.1 设计架构 5](#_Toc56430716)

[2.2 功能列表 5](#_Toc56430717)

[3. 数据流 6](#_Toc56430718)

[3.1 数据读取 6](#_Toc56430719)

[3.2 卷积计算 6](#_Toc56430720)

[3.3 激活、池化 6](#_Toc56430721)

[3.4 数据输出 6](#_Toc56430722)

[4. Testcase plan验证计划 7](#_Toc56430723)

[4.1 卷积层验证计划 7](#_Toc56430724)

[4.2 FC层验证计划 7](#_Toc56430725)

[4.3验证点 8](#_Toc56430726)

[5. Testbench验证平台 9](#_Toc56430727)

[5.1 验证语言 9](#_Toc56430728)

[5.2 验证工具 9](#_Toc56430729)

[5.3 验证目标 9](#_Toc56430730)

[附录 10](#_Toc56430731)

# 修改记录

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

版本：v1.0

时间：2020/11/13

修改人：黄俊光

修改内容：

编写高能效ECG检测CNN加速器验证报告初稿。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

版本：v2.0

时间：2020/11/15

修改人：李支青

修改内容：

添加数据流及卷积层验证点。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# 文档信息

## 1.1 功能模块信息

|  |  |
| --- | --- |
| 项目名称 | 基于量化卷积神经网络的高能效ECG检测加速器设计 |
| 功能模块 | ECG\_TOP |
| 设计人员 | 李支青、黄俊光 |

## 1.2 版本

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **日期** | **描述** | **作者** |
| V1.0 | 2020/11/13 | 完成ECG加速器功能验证 | 李支青、黄俊光 |
|  |  |  |  |

## 1.3 参考文档

|  |  |  |
| --- | --- | --- |
| **名称** | **日期** | **作者** |
| 高能效的ECG检测CNN加速器设计文档 | 2020/09/30 | 李支青、黄俊光 |
|  |  |  |

# 模块设计

## 2.1 设计架构

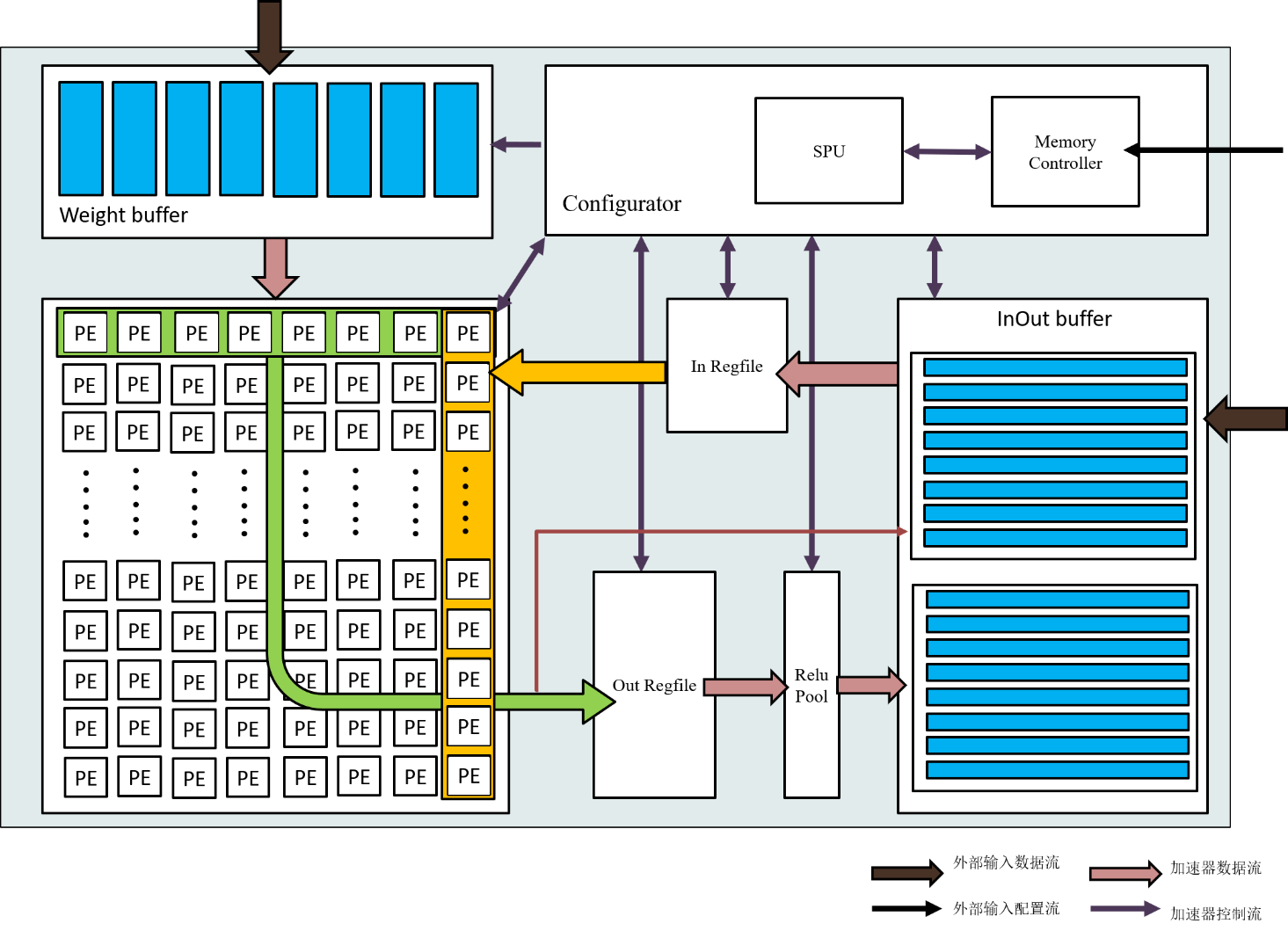


图2.1 ECG检测CNN加速器设计架构

## 2.2 功能列表

1. Inout buffer读写功能;
2. Weight buffer读写功能；
3. Input regfile读写功能；
4. PE阵列卷积计算功能；
5. Output regfile读写功能；
6. Relu & Pooling功能；
7. FC层功能；

# 数据流

## 数据读取

1. 使用python将ECG输出及网络权重数据导出为二进制mem文件；
2. In out buffer读取mem文件将数据保存在inoutbuffer的8个bank中；
3. Weight buffer读取权重mem文件，将数据保存在weight buffer的8个bank中；
4. In Regfile从Inoutbuffer中读取待计算数据，并将数据展开为16个输出；

## 卷积计算

1. In regfile将ECG数据对应输出至第一列PE计算单元（16个）；
2. 同时，weight buffer bank0将权重（同一权重）输出至第一列PE计算单元（16个）；
3. ECG数据与权重在PE单元中进行8bit乘法和16bit加法；
4. 下一时刻，各PE计算单元将上一时刻接收到的ECG数据传递至下一列PE计算单元，并接受上一列PE计算单元的数据输入（或第一列PE计算单元接收in regfile的数据输入）；
5. 完成单个输出的计算后，每列PE单元将计算结果截断为8bit，并传递至上一行对应的PE单元；
6. 所有的输出通过第一行的PE单元（8个）输出至out regfile；

## 激活、池化

1. PE阵列输出的数据将暂存在out regfile中，然后再以池化的大小和间隔输出数据到激活池化单元；
2. 然后进行最大池化，将池化的结果进行激活，再输出到inout buffer中。

## 数据输出

1. Relu\_Pooling之后的输出结果每次输出8个数据，然后按照地址的顺序写入inout buffer的bank0-bank7中；
2. 所有计算结果输出完毕之后，将inout buffer中的结果导出为二进制的men文件，以用于数据测试对比。

# Testcase plan验证计划

加速器整体验证过程分为2个部分，6层卷积层和1层FC层验证。其中最大的部分是卷积层的验证，通过python脚本可以将软件的计算结果和硬件的推理结果作对比，以便于找出有问题的模块和具体所在层数。

## 4.1 卷积层验证计划

卷积层一共有六层，每一层的验证流程相同，具体流程如下图所示。根据RP输出结果和卷积计算结果与软件结果对比，以此来划分问题所在的模块。



图4.1 卷积层验证流程图

## 4.2 FC层验证计划

FC层就只有一层，具体流程如下图所示。根据FC层卷积计算结果与软件结果对比，以此来划分问题所在的模块。



图4.2 FC层验证流程图

## 4.3验证点

|  |  |  |
| --- | --- | --- |
| **序号** | **验证点** | **进度** |
| 1 | ECG/权重数据输入及采样 | √ |
| 2 | PE计算单元乘累加计算 | √ |
| 3 | 相邻PE计算单元间的数据传输及采样 | √ |
| 4 | PE计算单元结果输出及Out regfile采样 |  |
| 5 | R&P数据输入及采样 |  |
| 6 | R&P数据处理 |  |
| 7 | R&P数据输出及InoutBuffer采样 | √ |
| 8 | 误差积累及传递 |  |
|  | 待补充 |  |

# Testbench验证平台

## 5.1 验证语言

Verilog、Python2.7

## 5.2 验证工具

VCS、DVE、Anaconda3

## 5.3 验证目标

RTL可以对十七种心率不齐的信号进行分类，并且分类准确率达到90%。

# 附录