Attention 模块 设计报告

张浩宇
522031910129

一、 模块设计

1.1 设计目标

设计一个模块进行 Self-Attention 计算，输入 Q、K、V 矩阵，得到输出结果，并进行量化。

模块参数设为：Token 数量为 8，Token 特征维度为 4，数据采用 Fix_8_8 即 8 位整数、8 位小数的定点无符号数格式存储和量化。

用 python 设计 golden model 用以验证，并用 verilog 完成硬件实现，要求结果仿真结果正确，代码可综合。

1.2 设计细节

1.2.1 Softmax 简化

标准的 Softmax 计算公式为：

$$\text{softmax}(A[i,j]) = \frac{\exp(A[i,j] - \max(A[:,j]))}{\sum_{i=1}^{N} \exp(A[i,j] - \max(A[:,j]))},$$

实现难度较高，因此采用以下简化的 Softmax 计算方法：

$$\text{softmax}(A[i,j]) = (A[i,j] - \min(A[:,j]))^2$$

1.2.2 数据量化

本设计采用Fix_8_8格式的16位定点数来存储数据，即8位整数、8位小数。但在Attention计算过程中，矩阵乘法和 Softmax 计算会导致数据位宽增大，因此在计算过程中需要将数据量化。

具体而言，在矩阵乘法中，经过乘累加运算，输出的数据为 Fix_18_16，需将其量化为 Fix_8_8；在 Softmax 计算中，经过平方运算，输出的数据为 Fix_16_16，需将其量化为 Fix_8_8。

量化过程包括两种情况。在小数位数不足以表达该数值的小数部分时，本设计采用舍入的方法，即根据超出部分的大小，若小于最小精度的一半，则舍去，否则进一位。在整数部分超出位数，发生溢出时，本设计采用饱和的方法，即近似到最大的正值。

1.3 Golden model 设计

用 python 设计该模块的 golden model，模拟计算中的矩阵乘法、softmax 和量化等过程，以验证硬件实现的正确性。为了编写方便，矩阵操作采用 pytorch 实现。

同时设计了 python 测试脚本，随机生成模块的输入数据，调用 modelsim 进行仿真，并将仿真结构与 golden model 计算结果比较，以验证硬件实现的正确性。

1.4 硬件模块设计

1.4.1 实现原理

Attention 计算可以分为三个步骤，分别是：矩阵乘法 $Q*K^T$、Softmax 计算、矩阵乘法 Score*V。本设计采用流水线结构实现，即每一个步骤作为流水线的一级，形成 3 级流水，3 个周期即可完成全部计算，如图 1。

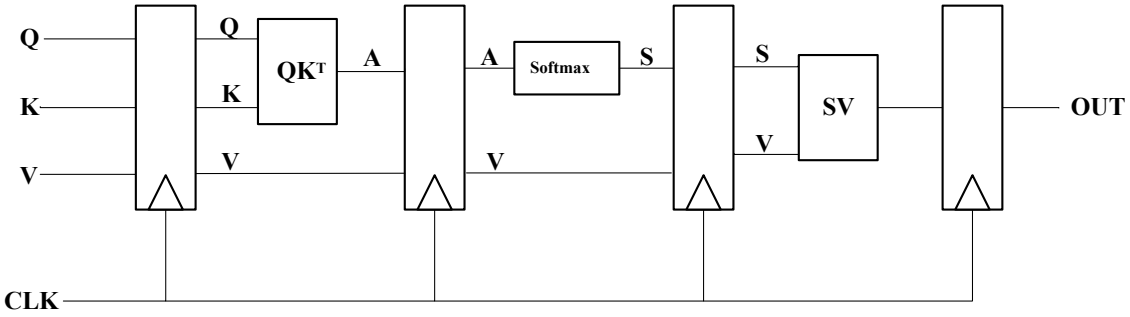

图 1　Attention 计算的流水线结构

1.4.2 模块总览

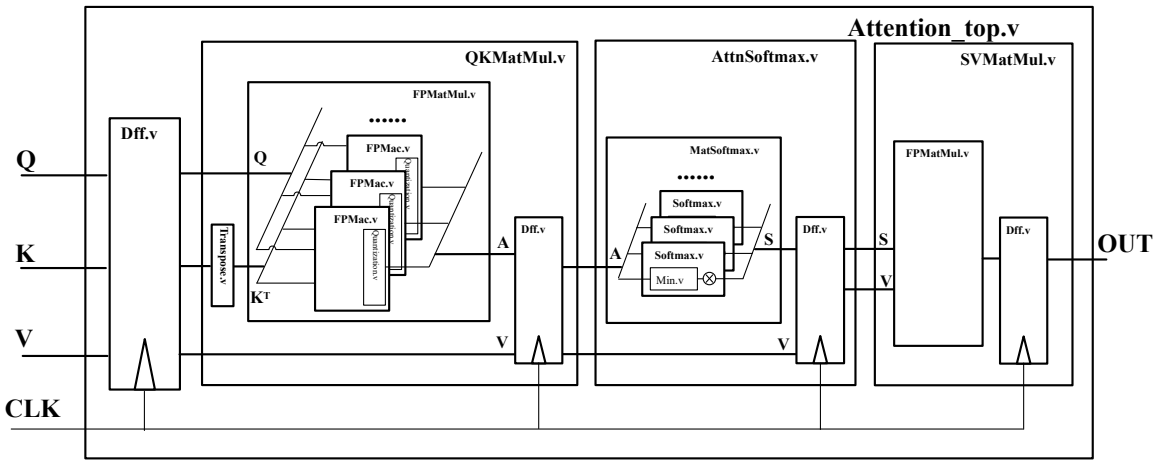

图 2　硬件模块设计

在模块设计中，各个子模块相关接口均采用参数化设计，便于模块复用与扩展。

1.4.3 量化模块

1.4.4 乘累加（MAC）模块

1.4.5 矩阵乘法模块

矩阵乘法实现两个输入矩阵的乘法运算和结果的量化。

1.4.6 转置模块

转置模块对输入矩阵进行转置，用于 $Q*K^T$ 的计算中。

1.4.7 Softmax 模块

1.4.8 寄存器模块

1.4.9 顶层模块与 testbench

二、 设计结果

2.1 计算结果

2.2 时序

2.3 综合

三、 讨论

3.1 数据流分析

3.1.1 关键路径

3.1.2 延迟与吞吐率

3.1.3 折叠设计

3.2 量化种的舍入