《SoC 设计方法学 project 指导书》

加速器设计

《SoC 设计方法学》课程结合 SoC 设计的整体流程，对 SoC 的含义、设计方法学以及如何实现进行了全面讲解，要求学生们在上完该课程后能对 SoC 的含义、架构和模块等知识有更系统的了解和掌握，有助于学生们从事 SoC 相关工作。在之前的课程学习和课程实验中，学生已经积累到了一定 SoC 的设计经验。本次 project 要求学生应用所学知识，进行加速器的所有硬件设计，其中包括代码设计、前端仿真、综合与后端设计等。

1 Project 简介和准备

1.1 Project 简介

请同学们自行分组，每组不多于三人，应用实验课上所学内容，应用硬件和软件代码知识，应用各种 EDA 工具，通过小组分工合作，共同完成**多头自注意力机制(MHSA)**的硬件加速设计，需要做的工作概括如下：

1、完成系统与加速器设计、验证方案撰写；

2、用Verilog语言完成RTL设计；

3、建立验证环境，编写testbench，验证加速器功能正确；

4、将加速器与蜂鸟E203连接，编写C程序让CPU控制加速器运行；

5、完成后端设计（后续实验课上会讲）。

1.2 Project 准备

Verilog/System Verilog 知识；

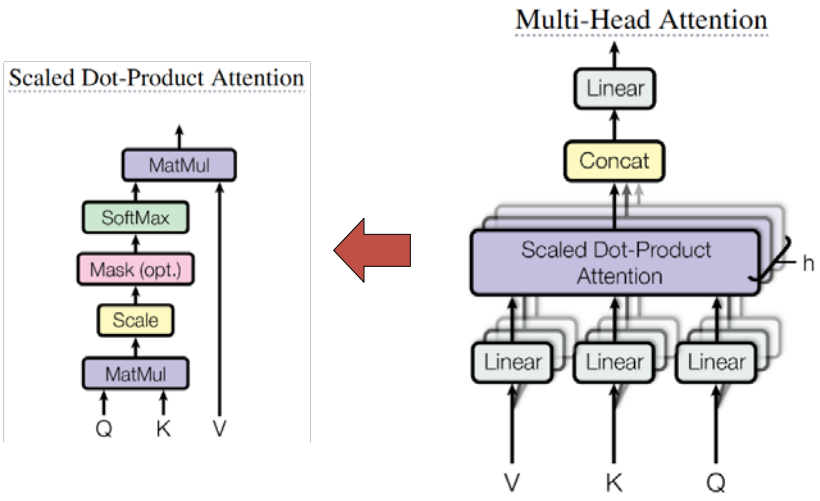
C/C++/Python 等语言知识；

所有 EDA 工具知识；

你的组员

2 理论基础

Google Brain 团队 2017 年的 Transformer 网络模型首次提出注意力机制。Transformer 多头自注意力机制的思想应用到了后续大量的网络模型中。关于注意力机制和 Transformer 的知识在网络上已有大量资料，请同学自行查阅并了解。需要实现加速的 MHSA 模块整体结构如下：



3 实验内容与实验要求

3.1 加速器功能说明

实现多头自注意力机制模块硬件加速，需要注意以下几点：

1.所有输入参数和输入权重至少为8比特有符号整数；

2.所有输入参数和输入权重都需要保存到外部存储中，设置好起始地址后可从 memory 读入；输出结果可以写入到 memory 中。

3.需要实现 Softmax 非线性层和 Scaling 层；至少需要实现四头注意力机制；输入特征维度（B，L，C）不得低于（1，32，128）；不可通过线性层收缩 Q、K、V 维度降低计算量，即每种向量在各头的通道数之和不得低于 128。

3.2 加速器功能限定

1. memory限定（如果需要）：课程提供的sram模型，sram有4k*64和8k*32两种规格；课程提供sram的仿真模型和综合用的库；提示：如果用课程提供的sram模型而自己又设计了过大的memory可能导致无法综合！

2. 乘法器限定：举例，若每个乘法器的输入为两个8比特有符号数，输出则为16比特有符号数；运用模块复用，总加速器不能用超过480/1920/19200个乘法器；（达到不同的设计指标会获得不同的难度分数）

3. 加速器优化目标：a.计算速度：综合时钟周期*总计算时钟周期数(包括搬运数据)；b.数据复用：与总线之间的数据传输量尽可能的少；c.运算单元复用：考虑加速器运行周期、加速器综合频率、加速器面积等因素权衡取舍，自行构建评价标准，建立你认为最优的加速器。

3.3 加速器功能验证

为了验证加速器功能正确性，需要编写一个Golden Model；

> “Golden Model is a model which represents some existing known good behavior of a function, normally created outside the scope of the verification activity. When one exists, it is useful in a verification environment to form part of a scoreboard / predictor or other checker arrangement to enable self-checking.”

Golden Model可以用任何语言编写（推荐用C、C++、Python或System Verilog）；Golden Model自身的正确性可以用现有框架（Pytorch等）来验证（Golden Model自身正确性验证不做强制要求）。在编写好Golden Model后请通过Verilog或System Verilog来编写testbench结合Golden Model来验证设计的功能正确性。

3.4 加速器的系统集成和功能验证

加速器运行规则：系统集成在蜂鸟E203 SoC环境下实现，需编写CPU可执行的C程序。E203具体资料链接：https://doc.nucleisys.com/hbirdv2/core/core.html

整体运算流程包括：

1. 所有输入存在总线可访问的地址空间中；

2. CPU通过读写加速器所在的地址空间来配置加速器（配置包括输入地址、输出地址、启动信号、完成信号）；

3. 加速器接收到CPU的配置后开始运算;

4. CPU接收到最后检查运算结果。

3.5 综合及后端设计

要求完成加速器的综合DC及后端ICC设计，在无时序违例的情况下，综合到尽可能高的频率，减小加速器面积；当出现违例时，根据时序报告分析关键路径情况并修改设计代码或综合脚本，综合的频率与面积会影响到project最终分数。后端设

计硬性要求只有完成整个流程，修复时序使得建立、保持时间违例尽可能少。

3.6 报告要求

本Project随课程期间进行，按照设计流程，以小组为单位随课程提交报告。报告中要求对上述五个要求的实施方法及过程有详细的说明，project报告按照模板书写。报告中需要有小组内成员的分工及工作量说明，最后打分会根据各个同学负责的部分完成的情况分别打分。组内成员应根据各部分的复杂、难易程度均衡划分工作量！具体时间安排如下表所示：

报告内容	提交时间	提交内容
系统设计方案	第七周	提供设计方案，包括基本计算原理、加速计算原理，理论加速比、系统架构、加速器设计方案（Spec 文档，包括加速器架构、接口、数据流、Memory Map 等）
加速器验证方案设计	第九周	根据加速器设计方案提取功能点，并完成模块验证方案设计（应包含测试用例的各级 feature、用例方式、检查方式、覆盖方式、回归次数等）
加速器前端设计与功能验证	第十二周	提交前端设计与功能验证报告（加速器单元功能验证即可，应包含 Golden Model 的实现、加速器详细设计细节、测试用例的视线、验证结果与覆盖情况，推荐使用 UVM 验证方法）
系统集成、系统级验证与测试	第十五周	完成加速器的集成，提交系统功能验证报告（系统集成的视线细节、系统验证环境搭建、驱动加速器的 C 程序、验证结果）
综合、后端设计与分析	第十七周	完成综合与后端设计流程，提交相关分析报告（应包含物理设计流程说明、违例分析与优化过程、及最后 GDSII 截图）

学期末要求将下列文件：加速器设计代码、加速器功能验证模型、SoC软硬件代码、综合脚本及报告、物理设计脚本（可选）、GDSII文件打包成压缩包并以小组为单位上交至canvas系统。

4 参考资料

[1] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [C]// Advances in NeuralInformation Processing Systems. 2017: 5998-6008.