AACS Project Report

023039910034 徐浏凯

实验要求

本次课程项目要求设计时序逻辑神经网络加速器，基于实验1中所设计的乘累加单元，实现特定功能的神经网络推理。要求所设计神经网络加速器能够在一定周期内输出结算结果，并进行DC综合，神经网络计算周期数不定，系统计算周期为10ns。本项目不要求系统吞吐、功耗、能效、延时等指标。

电路设计

神经网络算法分析

要求实现神经网络整体结构如下Fig. 1所示。整体网络中包含两个全连接层，两个scale层，和一个ReLU激活层。其中第一个全连接层执行两个尺寸分别为[1,100]和[100,10]的矩阵乘法，第二个全连接层执行两个尺寸分别为[1,10]和[10,10]的矩阵乘法。该神经网络能够实现对手写数字的识别和分类。

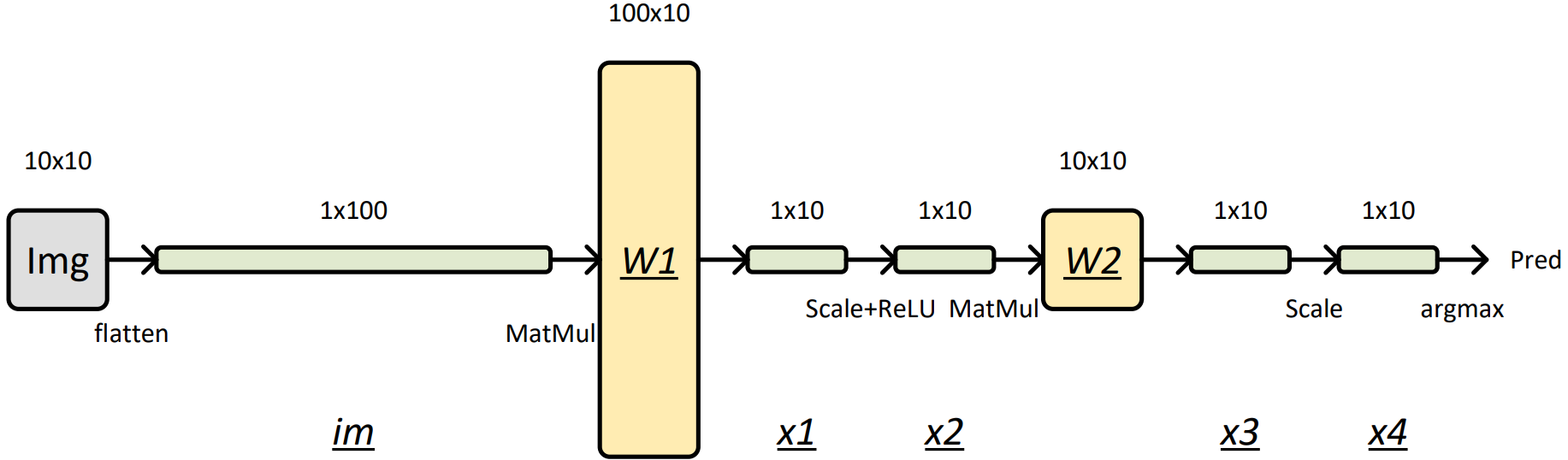


Figure 1 神经网络整体结构图

神经网络输入数据为10\*10的手写数字图片，以矩阵的形式输入，其中只包含0/1值，有笔记的位置为1，而无笔记的位置为0。Fig. 2分别展示了python代码中两张输入图像6和7的示例。

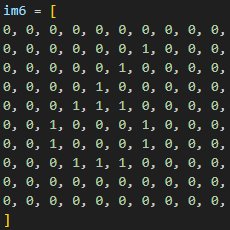
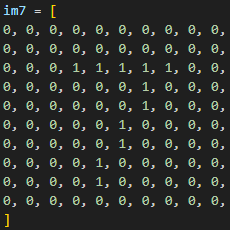
 

Figure 2 输入图像6和7示例

神经网络详细计算流程如下。首先，输入数据在进行计算前将先展开为1维形式，得到[1,100]的矩阵，再输入到第一个全连接层中，得到尺寸为[1,10]的输出矩阵。该输出矩阵将经过scale移位和ReLU激活层，得到尺寸为[1,10]的输出矩阵。该矩阵将输入到第二个全连接层，得到尺寸为[1,10]的输出矩阵，最后取得输出矩阵中最大值所对应的index即为神经网络的最终分类结果。综上所述，该神经网络的整体计算流程可以由Fig.3所示的公式进行表示。

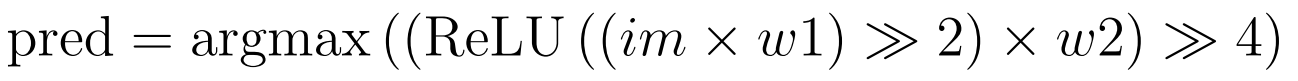


Figure 3 神经网络整体计算表示公式

其中，神经网络中的scale层以数据移位的形式实现，ReLU函数可由如Fig. 4所示的公式进行表示。在神经网络加速器中，所有的网络层的输入输出数据均以8比特有符号数的形式进行表示。

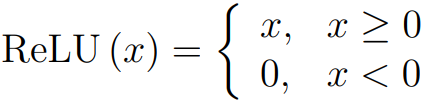


Figure 4 ReLU函数表示公式

另一方面，由于第一层全连接层的输入数据值仅为0/1，因此对于第一个全连接层，利用实验1中的可重构乘累加单元，能够在不损失神经网络精度的前提下实现两张输入图像同时推理。

神经网络加速器电路设计方法

（一）数据流控制

由于整体神经网络包含多个层，因此通过状态机对整体神经网络执行流程进行控制。设置state变量控制整体计算流程，其整体变化流程如下图所示。整体流程上，神经网络逐层执行，因此状态机也设置为顺序跳转，在一张图片执行完成后，根据预先配置的并行执行寄存器，直接跳出计算回到IDLE状态或继续执行第二张图片的后续计算。



Figure 5 状态机STATE变量跳转图

实际RTL代码根据状态机STATE变量变化图进行撰写，由于代码段较长因此不在这里贴出完整代码。所设计加速器在接收到start信号后直接开始计算第一个神经网络层，并在后续过程中通过case语句进行判断和状态跳转。在完成神经网络整体计算流程后加速器将重新回到4’b0000的IDLE状态。

（二）加速器单元设计

仿真结果

通过vivado工具对电路进行仿真测试，以输入30°和45°角作为激励，验证电路单元的功能正确性，具体testbench代码见附录。仿真时间长度为400ns，每200ns进行一次输入。输出波形结果如下图所示。

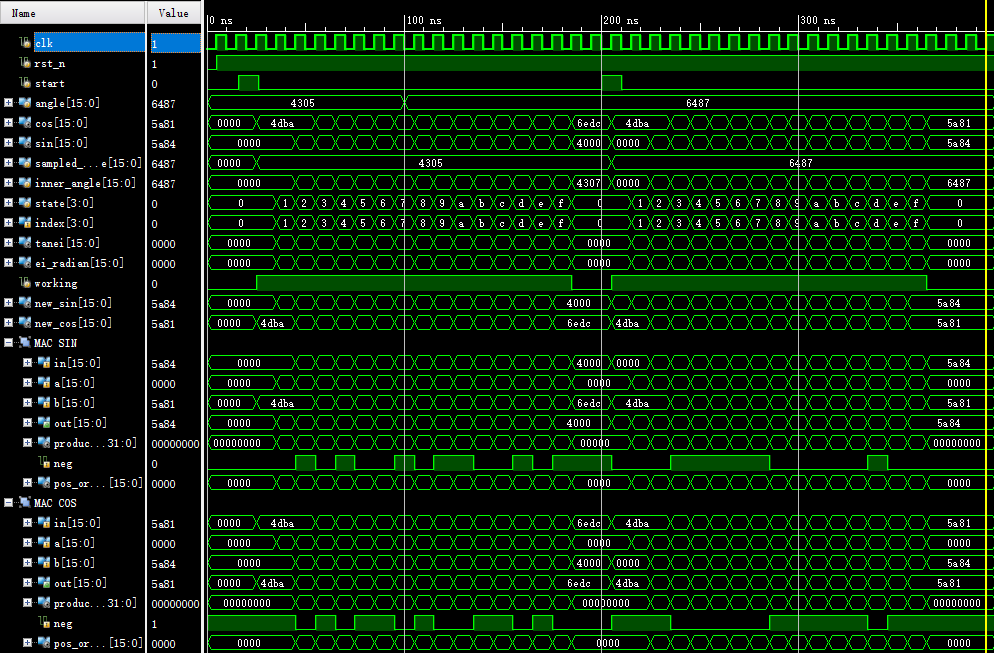


Figure 7 Vivado仿真输出波形结果（30°和45°）

从图中可以看到，所编写的电路单元能够有效支持CORDIC算法的计算，能够正确输出对应sin值和cos值，完整计算延时为16个周期，即160ns。Working信号为工作信号，当该信号拉高时，表示正在计算，该信号被拉低后的输出才是最终的sin值和cos值的计算结果。

其中当输入为30°时，最终输出sin值为16‘h4000，输出cos值为16‘h6edc，考虑对应数据包含1位的整数位和15位的小数位，sin值16‘h4000和cos值16‘h6edc分别代表0.5和0.866089，相比精确值的误差分别为0.00000%和0.00734%。当输入为45°时，最终输出sin值为16‘h5a84，输出cos值为16‘h5a81，考虑对应数据包含1位的整数位和15位的小数位，sin值16‘h5a84和cos值16‘h5a81分别代表0.707153和0707062，相比精确值的误差分别为0.00654%和-0.00633%。

另一方面，观察中间数据结果，以30°计算过程为例，如下图所示。可以看到，在计算过程中，sin值和cos值逐渐趋近于最终的计算结果16‘h4000和16‘h6edc。随着计算过程中时钟周期的不断推进，最终输出计算结果sin值为精确结果，cos值最终输出结果为各个周期中误差结果最小的数据。但此时cos并不是16比特数据中最接近理想值的，最靠近理想值的是16‘h6eda，若继续迭代更多计算周期，同时提高LUT的数据表示位宽，可以使最终计算结果更趋近于理想值。

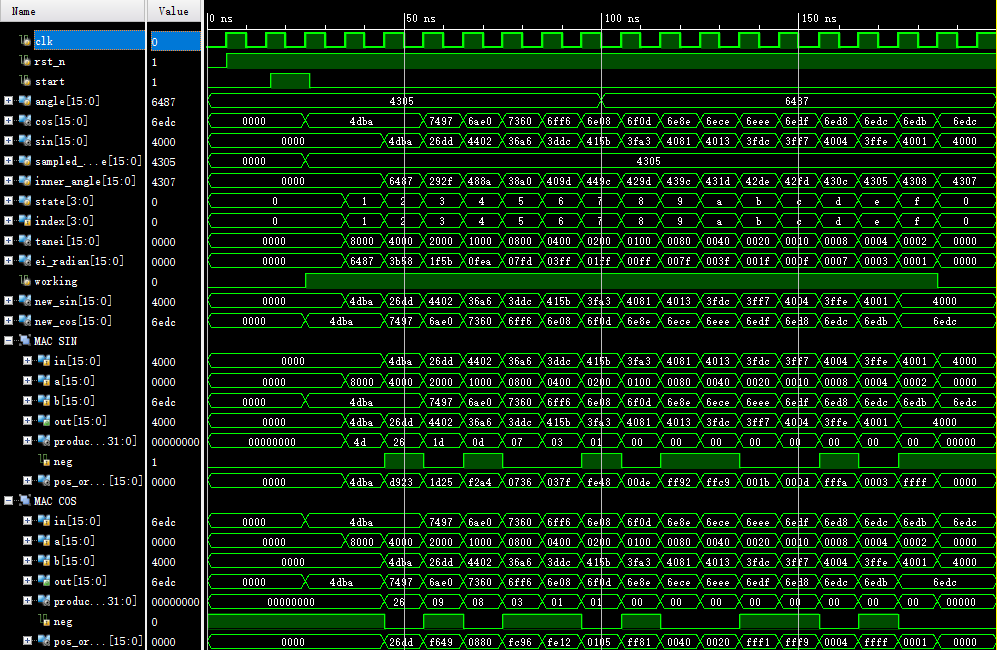


Figure 8 vivado仿真输出详细波形结果（30°）

DC综合结果

对所编写的电路通过DC工具进行综合，功耗和面积结果如下表所示。详细DC输出文件见附件。

Table 2 Power and area of the proposed CORDIC unit

|  |  |  |
| --- | --- | --- |
|  | Power (uW) | Area (μm2) |
| Proposed CORDIC unit | 134.4940 | 2336.983098 |

由表可知，所设计的CORDIC计算单元功耗为134.4940uW，面积为2336.983098um2。其中，功耗部分，寄存器功耗为74.2504uW（占比55.21%），组合逻辑功耗为60.2436uW (占比44.79%)。面积部分，组合逻辑面积为1335.272581 um2，缓冲单元面积为44.221056 um2，非组合逻辑部分面积为491.006217 um2，互连线部分面积为510.704301 um2。

在时序上，输出slack为6.03ns，有足够的裕量进行计算。

Discussion

在单元设计之初曾考虑通过实时监测向量的变化，并在每一个周期都控制向量模长为1的方式。这种方式能够在输入的角度值正好等于部分角度组合的情况下提前输出计算结果，整体计算周期数不固定，但是这种情况在所有角度中的占比非常小，因此最后没有考虑这种方案，采用了固定周期数输出的方案。相比不固定周期数的输出方案，固定周期数的方案不需要每个周期保持输出向量模长相等，因此也不需要额外的LUT来输出每个周期不同的K值，能够在一定程度上降低整体电路的面积。

Appendix

Testbench top\_tb.v

accelerator.area.rpts

accelerator.power.rpts

accelerator.timing.rpts