实验步骤

1. 编写LSTM IP核

本实验提供了LSTM IP核的HLS模板工程。模板工程共含有3个源文件和3个头文件,如表2-1所示。 表2-1 HLS模板工程的源文件和头文件

源文件/头文件	备注
weight.h	存放已训练好的RNN网络的权值和偏置
utils.h	将RNN的输入/输出转换为总线接口格式的数据流
rnn.h	RNN网络的头文件
rnn.cpp	RNN网络的实现
rnn_top.cpp	HLS顶层函数
main.cpp	TestBench

同学们在实验时,只需补全 rnn.cpp 或修改 rnn.h 的相关代码即可,其余源文件和头文件可直接使用。

编写LSTM IP核的具体步骤为:

Step1: 补全LSTM IP核关键代码

打开实验包lstm_hls目录中的HLS工程,根据LSTM Cell的原理图和源文件中的代码注释提示,补全 rnn.cpp 中的LSTM前向推导相关代码。

Step2: 对LSTM IP核进行C仿真/CSim

打开 rnn.h 的头文件,将 CSIM_ON 的宏从0修改为1,从而打开仿真开关,如图2-1所示。

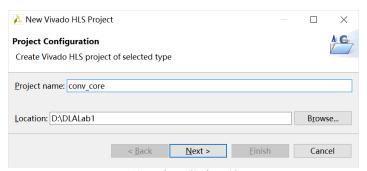


图2-1 打开仿真开关

点击HLS工具栏的 按钮,开始C仿真。此时,TestBench将调用rnn.cpp中的infer函数对MNIST的测试集数据进行测试。

如果编写的LSTM IP核功能正确,则仿真输出将形如图2-2所示。

图2-2 仿真输出结果参考

由图2-2可知,测试集的预测准确率为98.77%。

Step3: 综合/Synthesis

点击HLS工具栏的 按钮,对编写的RNN加速器进行综合,生成RTL电路。

综合成功后、可查看HLS生成的综合报告、必要时需要根据综合报告对IP核代码进行优化和改写。

!!! warning "注意!!! 📢"

在综合前,必须将 rnn.h 头文件中的 CSIM_ON 宏关闭。

Step4: 打包IP核

点击HLS工具栏的中按钮,对综合后的RTL电路进行打包,生成IP核。

2. 构建Block Design

Step1:新建Vivado工程

打开Vivado,选择正确的器件型号,新建名为mnist_lstm的工程。

Step2: 导入LSTM IP核

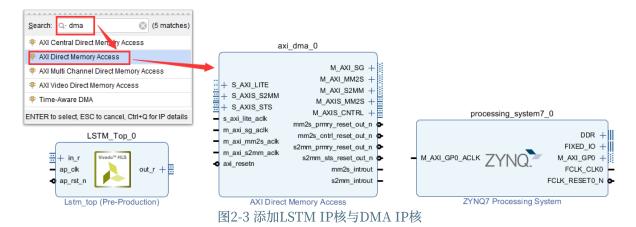
在工程设置项中, 把生成的LSTM IP核导入到当前的工程当中。

Step3: 新建Block Design

点击 "Create Block Design", 创建Block Design电路模块图。

Step4:添加IP核

在Block Design中,依次添加PS IP核、LSTM IP核与DMA控制器IP核,如图2-3所示。



Step5: 配置IP核

首先对PS的ZYNQ IP核进行配置:

点击"Run Block Automation"完成自动配置后,打开ZYNQ IP核的配置窗口,添加S_AXI_HP0总线接口,用于实现与LSTM IP核的数据交互。

然后对DMA控制器的IP核进行配置:

打开DMA控制器IP核的配置窗口、按图2-4设置DMA控制器。

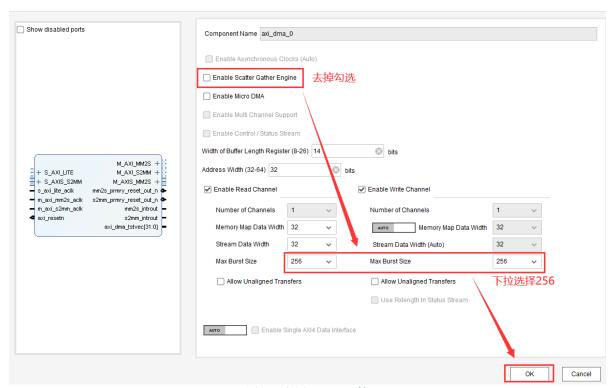


图2-4 配置DMA IP核

Step6: 连接IP核

点击 "Run Connection Automation", 勾选全部复选框, 如图2-5所示。

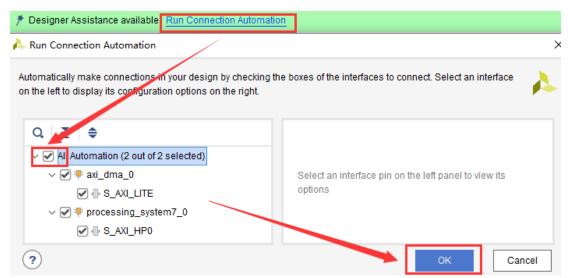


图2-5 自动连接所有IP核

此时,Vivado将尝试自动连接所有已添加的IP核。连接完成后,点击"Diagram"工具栏的C按钮以重新布局电路。

点击LSTM IP核的"in_r"接口,将其与DMA IP核的"M_AXIS_MM2S"接口连接起来,如图2-6所示。

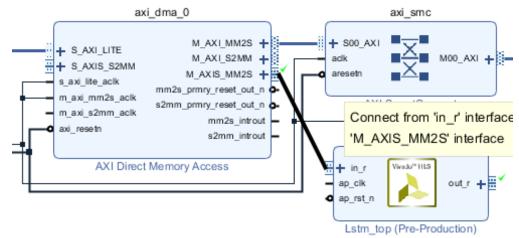


图2-6 将LSTM IP核的数据输入接口与DMA IP核连接起来

类似地,将LSTM IP核的"out_r"接口与DMA IP核的"S_AXIS_S2MM"接口连接起来。

连接完成后,再次点击"Run Connection Automation"以完成剩余线路的连接。

点击保存按钮保存Block Design。至此,Block Design构建完毕。

Step7: 生成比特流

点击"Create HDL Wrapper"以创建Block Design的HDL顶层封装文件,然后点击Vivado工具栏的显特短以生成比特流。

Step8: 导出Overlay

比特流生成完毕后,将.bit拷贝出来,重命名为mnist_lstm.bit。

此外,打开Block Design电路图,导出Overlay所需的.tcl脚本文件,并重命名为mnist_lstm.tcl。

将mnist_lstm.bit和mnist_lstm.tcl拷贝到实验包的lstm_app.zip中。

3. 上板测试

将实验包中的 lstm_app.zip 上传到Jupyter并在解压。

随后点击进入 lstm_app 目录。点击打开 mnist_lstm.ipynb ,点击Cell->Run All进行测试。观察 RNN的软硬件推导结果和推导时间的差别,如图2-7、图2-8所示。

Image labels: [40265769]



Result: 4, time: 0.391547s Result: 4, time: 0.281442s Result: 0, time: 0.388240s Result: 0, time: 0.281387s Result: 2, time: 0.388337s Result: 2, time: 0.281379s Result: 6, time: 0.387407s Result: 6, time: 0.281369s Result: 5, time: 0.395550s Result: 5, time: 0.281411s Result: 7, time: 0.387504s Result: 7, time: 0.281349s Result: 6, time: 0.388268s Result: 6, time: 0.281389s Result: 9, time: 0.387539s Result: 9, time: 0.281372s

(a) 软件运行时间 (b) 硬件运行时间 图2-8 观察软硬件推导的运行结果差异