基于PYNQ的HLS版本CNN卷积神经网络硬件架构

1. 硬件架构总图

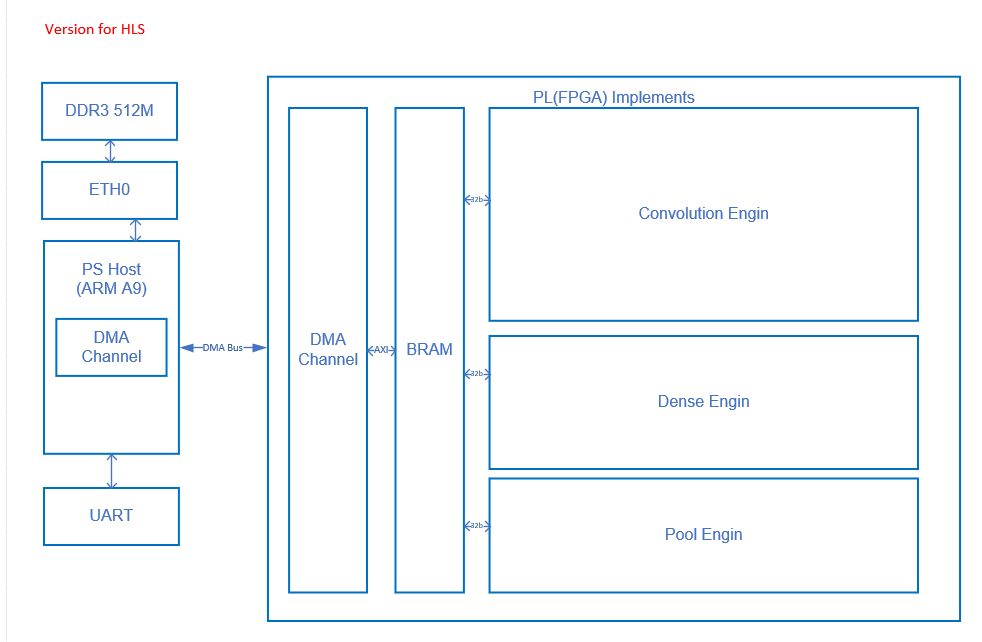


图1. HLS版本卷积神经网络硬件架构

如图1所示为HLS版本的卷积神经网络硬件架构框图，Verilog版本将在后期陆续推出。首先熟悉Lenet5 网络模型的人都知道它基本运算层就是卷积层（Convolution）、全连接层（Full connection or Dense）和池化层（Pool）, 这里的卷积引擎是一个可配置的卷积计算单元，能实现不同卷积核大小、尺度的计算，因此Lenet5中的C1,C2层卷积都可以通过它来实现，采用分时复用结构，同理FC1,FC2都是复用Dense Engin物理单元来实现计算，以及池化层，此外在FPGA中一般不会去实现softmax层，softmax主要作用是将全连接层的输出数据归一化，使之归一到0到1的范围内，就FPGA的资源而言若是通过Verilog语言编写这样的函数是非常困难的，当然在HLS中是容易实现的，这里我们可以将softmax层的执行放在ARM端，下面将逐一介绍工作原理。

当你在pynq 或 zynq 7000 系列成功移植前面所讲的PYNQ系统后，我们可以直接将数据集存放在SD卡上，当网络运行时，首先通过ARM端的app将SD卡的数据集加载到DDR中，然后分成若干BATCH将其通过DMA总线下发到图中的PL侧，PL侧使用一定大小来存放一张图片的所有像素信息，这里为28\*28大小的图像，此外BRAM还要存储网络中的权值参数和中间运算结果。使用HLS的好处就是直接可以将其定义为数组就可以自动分配相应大小的BRAM资源，然后将图像数据送入卷积引擎中计算，卷积计算完毕后会回写到BRAM中，然后再送入池化层和全连接层计算。计算的顺序根据Lenet5的网络顺序来实现。由于PYNQ-Z2上的BRAM资源有限，因此每计算一层网络都会将数据回写到DDR3内存中，接着重写配置下一个物理引擎的参数，然后开始将刚才的数据送入到PL侧开始计算。直至计算完毕。