

第九届

全国大学生集成电路创新创业大赛

报告类型： 仿真验证报告

参赛杯赛： 中科芯杯

作品名称：多精度张量计算单元

队伍编号： CICC0907607

团队名称： Ordovician

**目录**

1.对多种矩阵格式的支持展示

1.1 对m8n32k16 格式的支持展示

1.2 对m16n16k16 格式的支持展示

1.3 对 m32n8k16 格式的支持展示

2. 对多种数据类型的支持展示

2.1 对INT4 的支持展示

2.2 对INT8 的支持展示

2.3 对 FP16 的支持展示

2.4 对 FP32 的支持展示

3. 对特殊情况的支持展示

3.1 对混合精度计算模式的支持展示

3.2 对计算结果溢出情况的支持展示

3.2.1 对整形的计算结果溢出的支持展示

3.2.2 对浮点形的计算结果溢出的支持展示

4. 设计模块的FPGA验证展示

**快速预览简介**

本项目的验证在Vivado 2024.2平台上进行，采用Simulation-Run Simulation-Run Behavioral Simulation工具验证模块的逻辑正确性。具体的，本小组在仿真阶段选择例化三个Vivado提供的带AXI4接口的Block Memory，记为A，B，C并将矩阵数据和寄存器配置先行存放在A，B两个BRAM中，最后经过计算后输出到C矩阵中，以此来实现对整个模块逻辑的整体验证。

本文中，本小组将展示根据官方提供数据以及自拟的混合精度计算及溢出数据对本小组设计的模块的仿真验证，行文顺序将按照目录顺序展开。其中，对不同格式的矩阵形状的支持的验证都会在INT4数据格式下展开，对不同数据类型的计算支持的验证会在m16n16k16矩阵形状下展开，对混合精度的计算支持的验证是A,B矩阵采取FP16精度，C矩阵采取FP32精度的数据精度下展开，设计模块的FPGA验证采用的是Artix-7系列的xc7a100ticsg324-1L芯片进行验证，具体方法是把C矩阵的数据按16进制展示到数码管上，一次展示8位16进制数。

**1. 对多种矩阵格式的支持展示**

1.1 对m8n32k16 格式的支持展示

1.1.1 A，B存储器的填入数据展示

在这种情况下，填入A存储器的数据为00000000，表示A,B,C矩阵的精度都是INT4，矩阵格式是m8n32k16，填入B存储器的数据如图所示：（按16进制8位数行优先存储。矩阵数据连续存储，所有数据之间无空位无补零。）

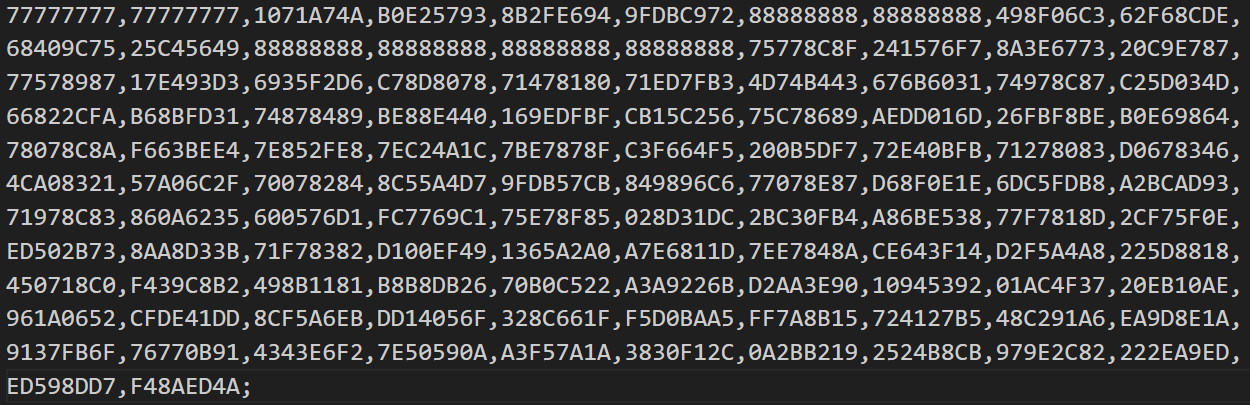


图 1：m8n32k16 格式的矩阵数据

1.1.2 经过读模块后预处理得到的数据展示

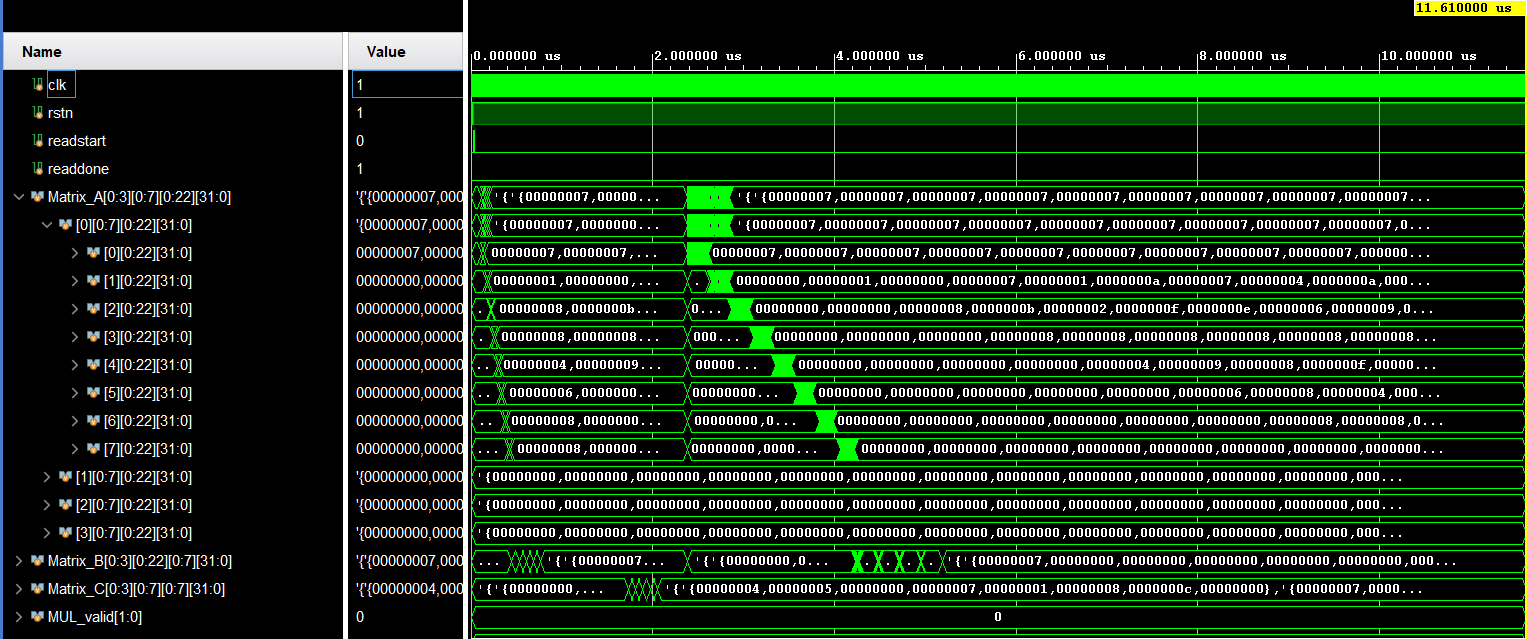


图 2：读模块得到的A,B,C矩阵数据

具体的寄存器设计详见设计报告，这里以A矩阵（8\*16）为例，可以在图中看见A矩阵已经被正确读入并预处理。具体的，A矩阵的第一块（[0:7][0:15]）被正确填入，并且按照阶梯形式加入了空白气泡供后面的脉动阵列计算使用。

1.1.3 计算结果展示

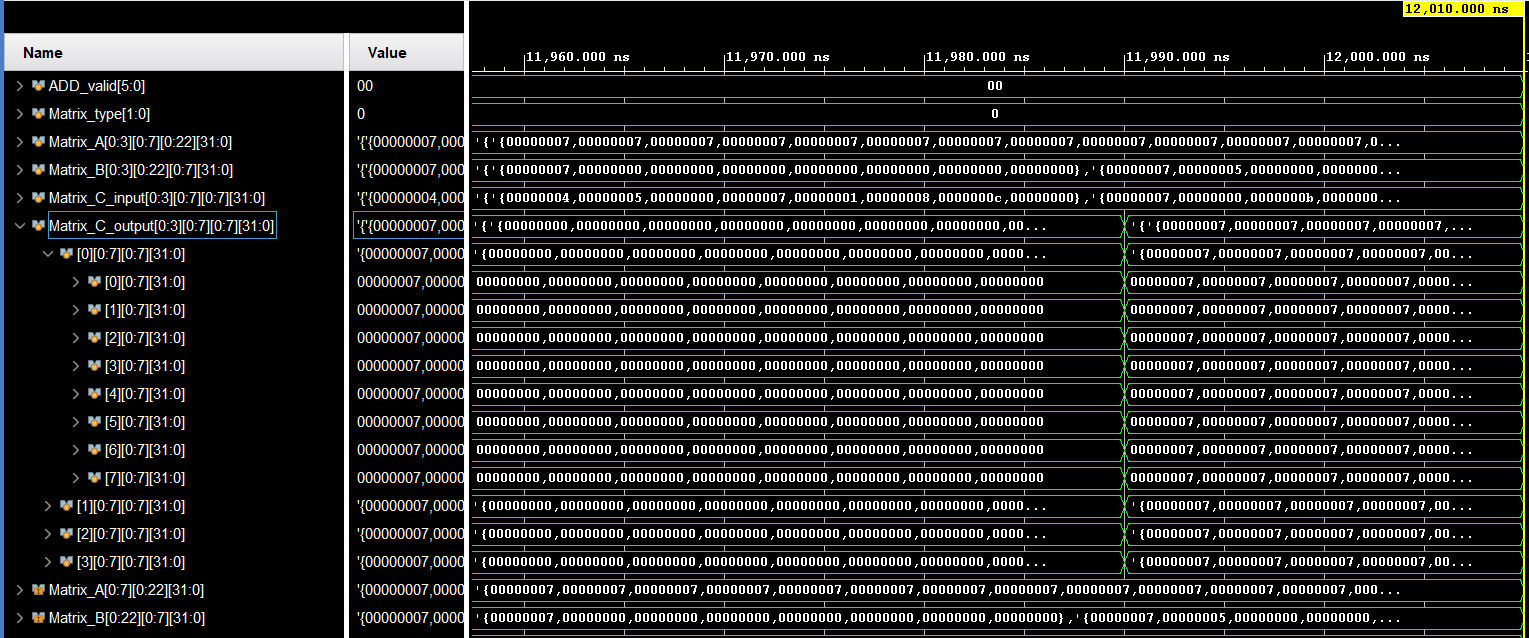


图 3：输出矩阵最终结果

可以看到，最终的结果都是7，这是为什么呢？因为最后的结果都溢出了，而我们小组设计的模块对整形的溢出采取饱和处理——及超过最大值的结果统一取最大值，小于最小值的结果统一取最小值。

1.2 对m16n16k16 格式的支持展示

1.2.1 A，B存储器的填入数据展示

在这种情况下，填入A矩阵的数据为00000001，表示A,B,C矩阵的精度都是INT4，矩阵格式是m16n16k16，填入B存储器的数据如图所示：

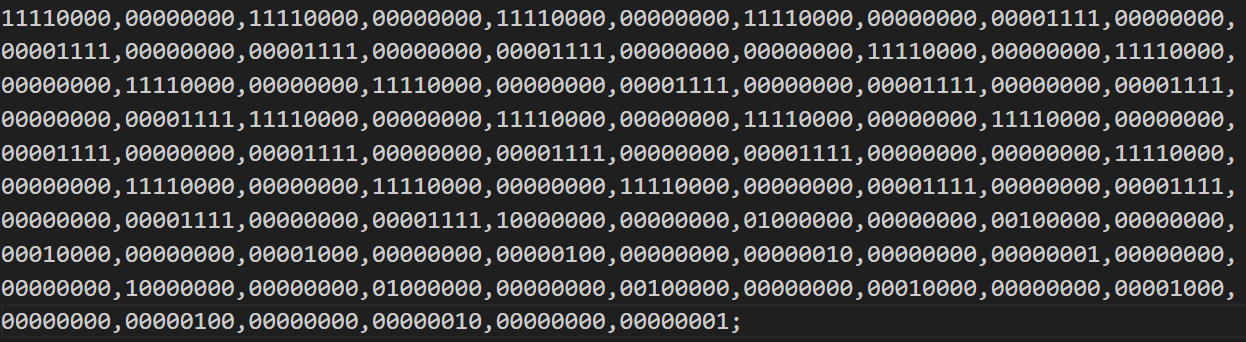


图 4：m16n16k16数据展示

1.2.2 经过读模块后预处理得到的数据展示

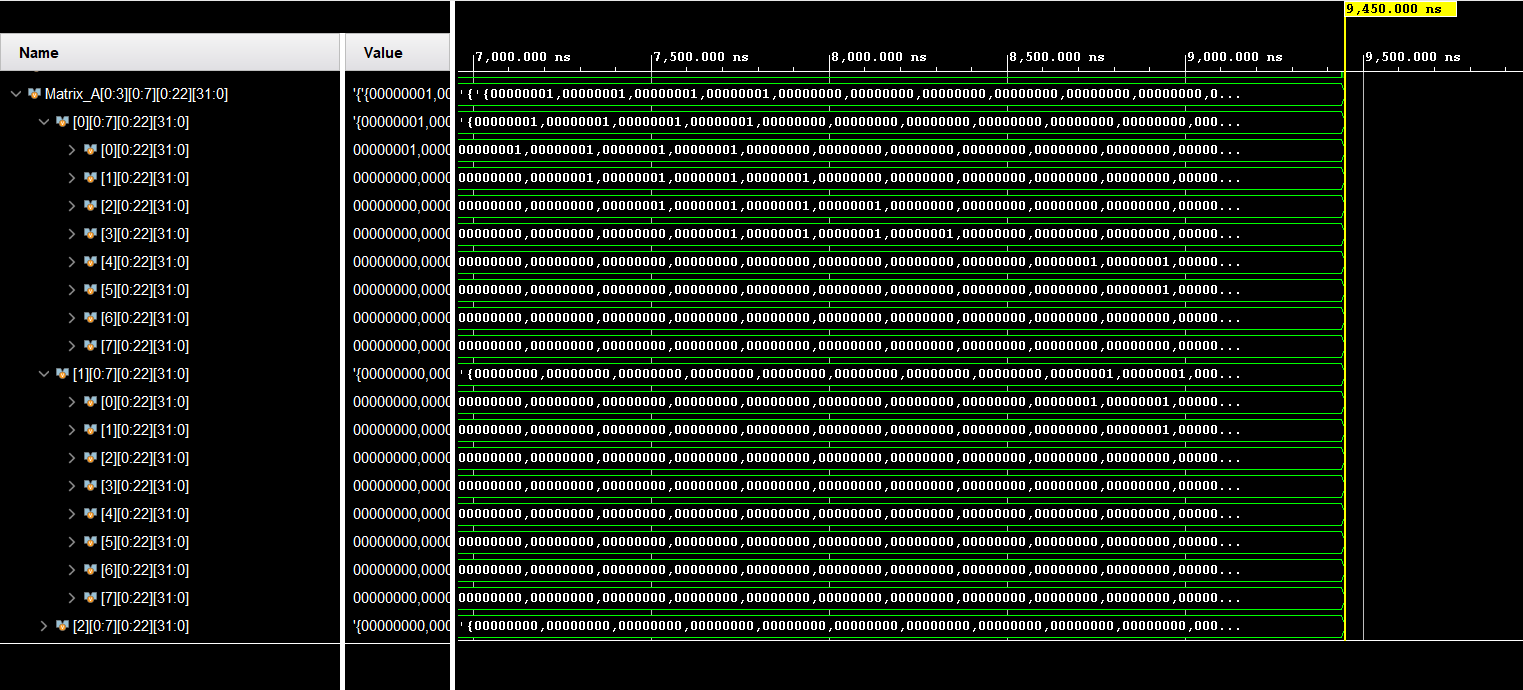


图 5:读模块得到的A矩阵数据

这里仅展示A矩阵的数据，A矩阵被分成两块[7:0][15:0]和[15:8][15:0],每一块独立的充入气泡供后面的脉动阵列处理。从图中可以看到，A矩阵经读入和预处理后的结果符合预期

1.2.3 计算结果展示

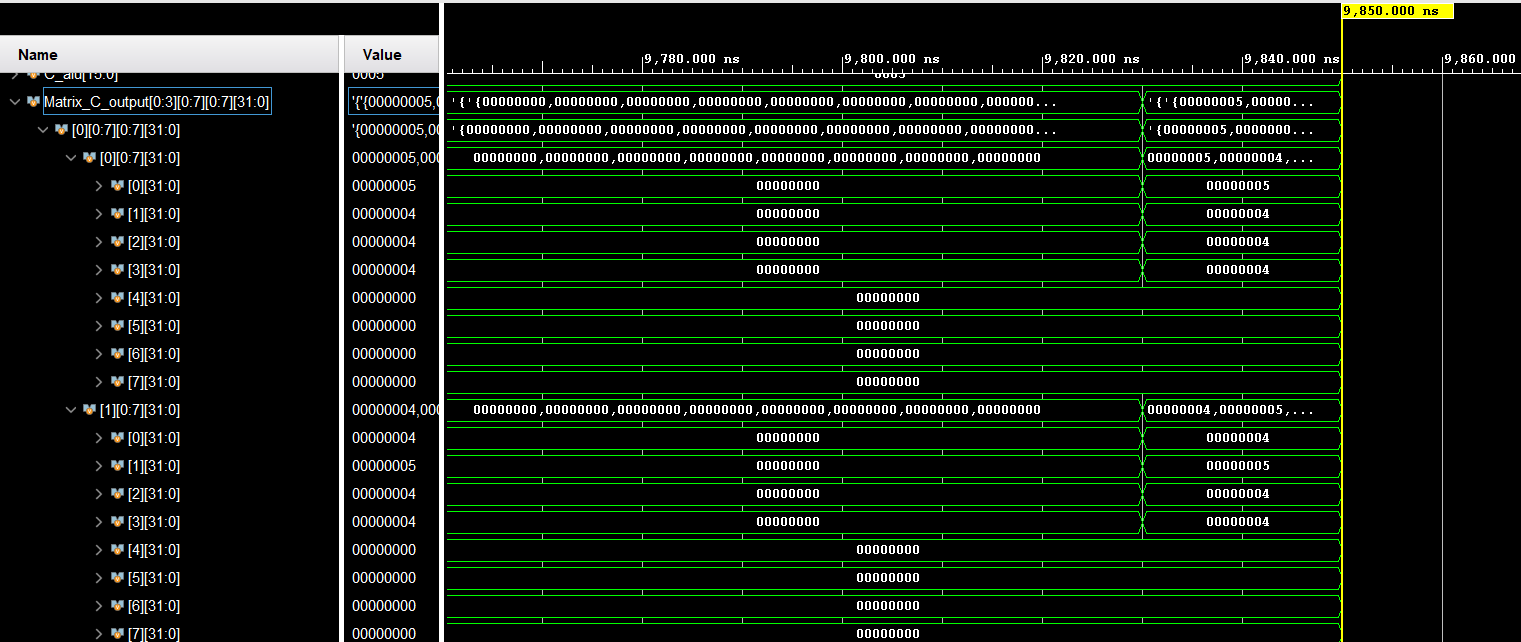


图6：输出矩阵最终结果

这里展示了输出矩阵的第一行第二行的最终结果，可以看到输出矩阵正确执行了A\*B+C的运算，以第一个数为例，A[0][0:15]={1,1,1,1,0,0…},B[0:15][0]={1,1,1,1,0…},C[0][0]=1,所以C\_OUT[0][0]=5是正确的。

1.3 对 m32n8k16 格式的支持展示

1.3.1 A，B存储器的填入数据展示

在这种情况下，填入A矩阵的数据为00000002，表示A,B,C矩阵的精度都是INT4，矩阵格式是m32n8k16，填入B存储器的数据如图所示：

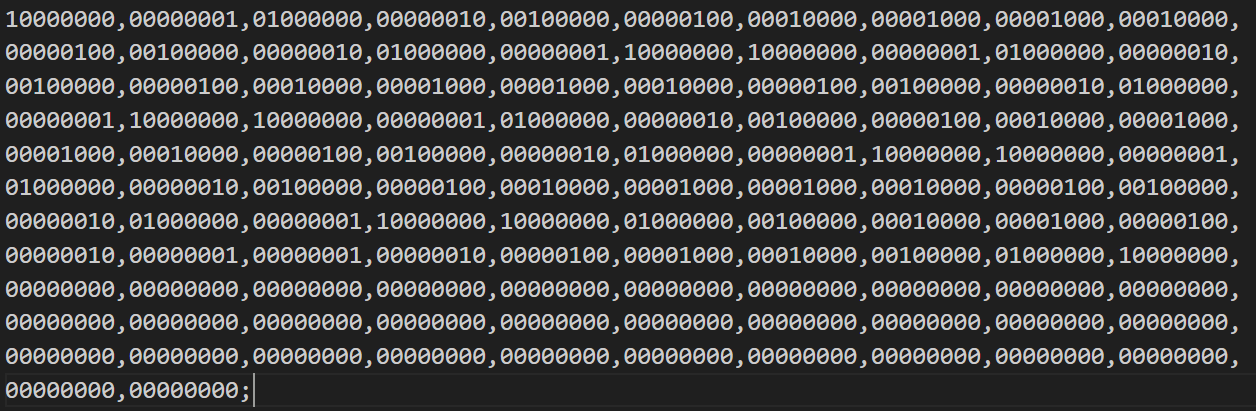


图 7：m32n8k16数据展示

1.3.2 经过读模块后预处理得到的数据展示

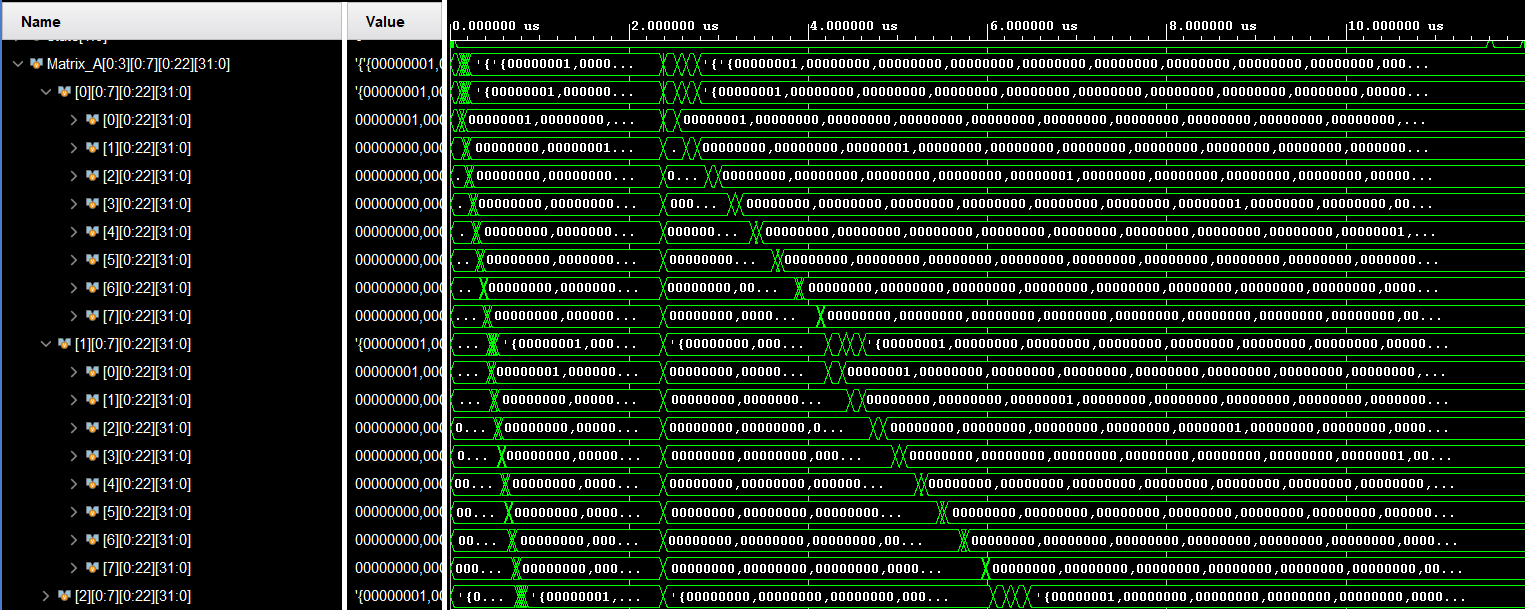


图 8：读模块得到的A矩阵数据

这里还是仅展示A矩阵的数据，A矩阵被分成4块，[7:0][15:0],[15:8][15:0],[23:16][15:0],[31:24][15:0],图中展示了一二两块，可以看到，A矩阵经过AXI总线读入和预处理后的数据是正确的，可以直接进入脉动阵列的运算

1.3.3 计算结果展示

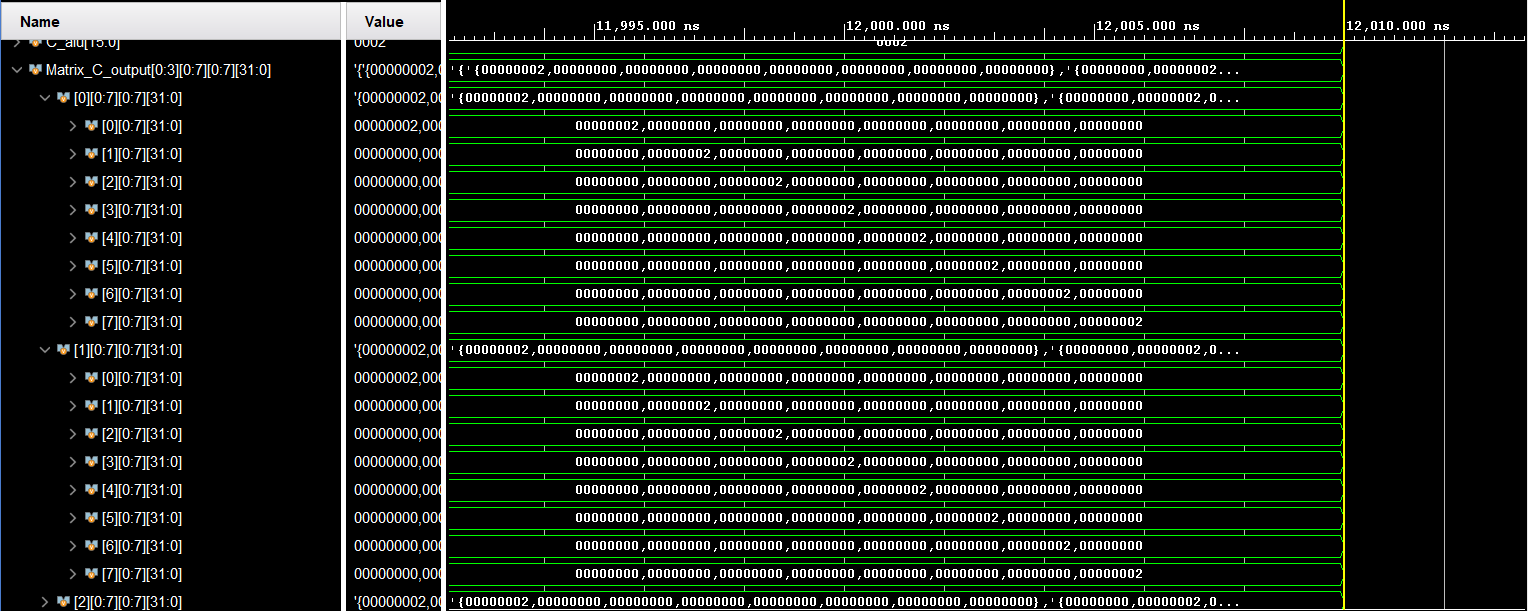


图9：输出矩阵最终结果

可以看到，输出矩阵最终每一块都是对角元为2的对角矩阵，符合输入数据应该得到的结果，举例来说，A[0][0:15]={1,0,0…,1},B[0:15][0]={1,0,0…,1},C[0][0]=0,所以C\_OUT[0][0]=2是正确的。

**2. 对多种数据类型的支持展示**

2.1 对INT4 的支持展示

由于上一节三个例子都是基于INT4的操作，故这里不再赘述

2.2 对INT8的支持展示

2.2.1 A，B存储器的填入数据展示

在这种情况下，填入A矩阵的数据为00000015，表示A,B,C矩阵的精度都是INT8，矩阵格式是m16n16k16，填入B存储器的数据如图所示：

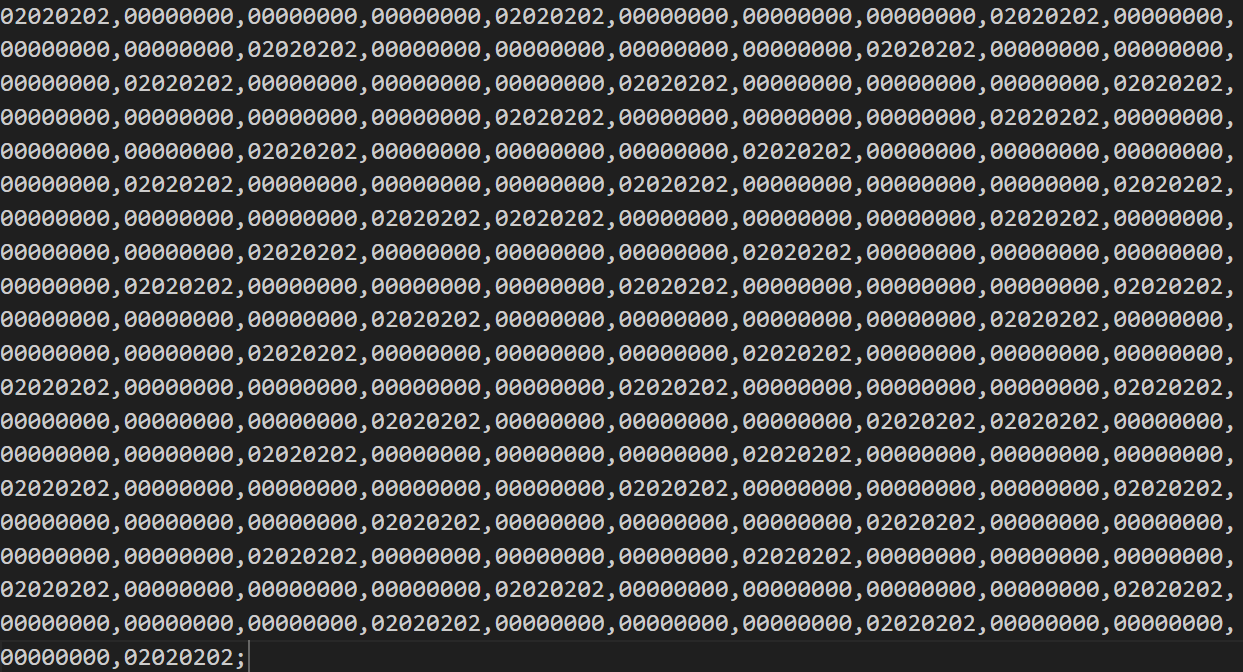


图 10：INT8数据展示

2.2.2 经过读模块后预处理得到的数据展示

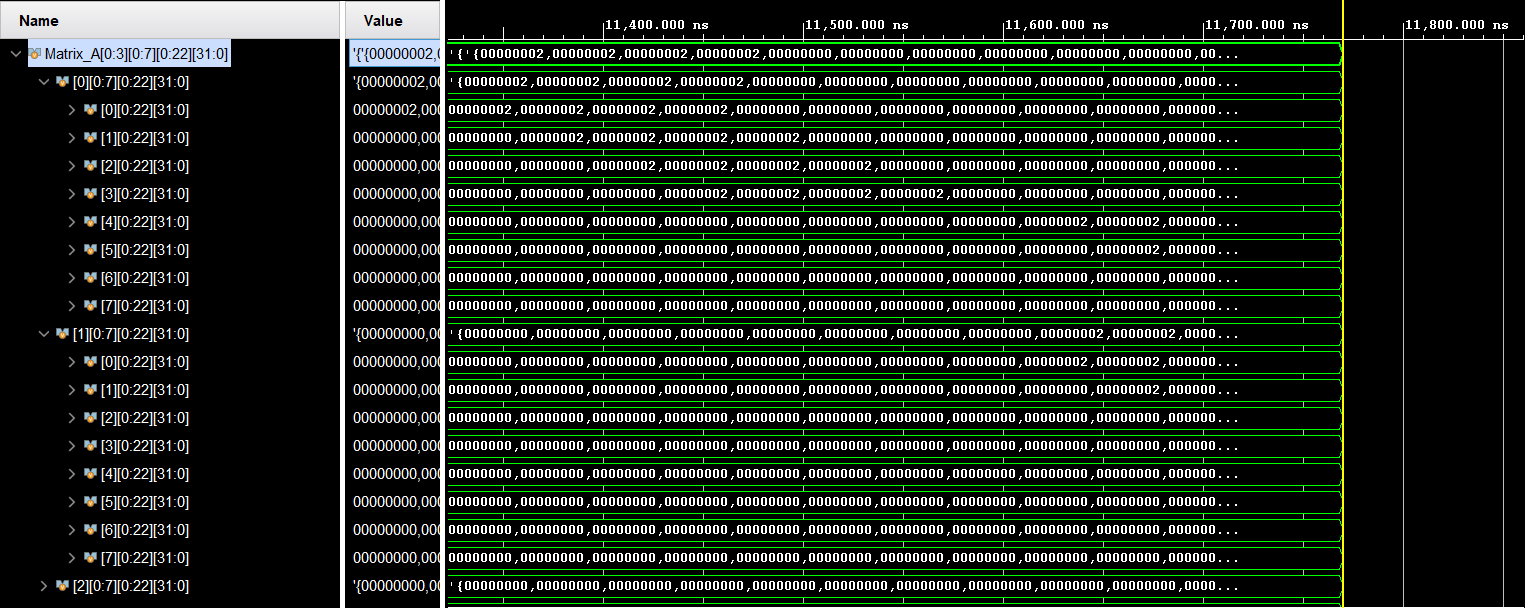


图 11：读模块得到的A模块数据

这里仅展示A模块的数据，如前所述，A模块被分成两块[7:0][15:0]和[15:8][15:0]。从图中可以看出INT8类型数据被正确读入并解析，即输入数据的每两个16进制数被解析成一个INT8数。然后矩阵被正确充入气泡以供后面的脉动阵列使用。

2.2.3计算结果展示

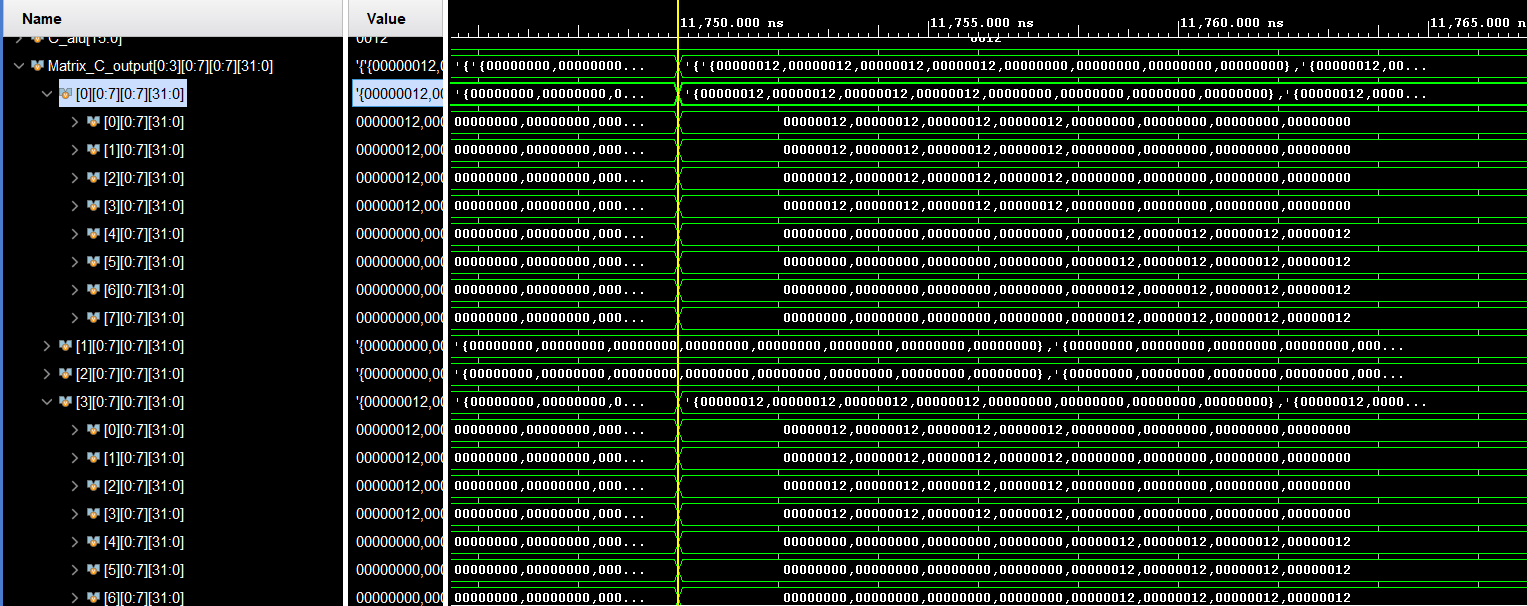


图12：输出矩阵最终结果

可以看到，最终的输出矩阵符合预期，举例来说，A[0][0:15]={2，2，2，2，0，0…},B[0:15][0]={2,2,2,2,0,0,…},C[0][0]=2,所以C\_OUT[0][0]=18=0x12是正确的。