矩阵乘法

1.矩阵乘法该怎么算？

FPGA实现矩阵快速相乘&矩阵求伪逆 - 知乎

看到一篇博客，对m*n $\times$ n*p 的矩阵乘法，可以在n个时钟周期内完成

但相应的，需要m*p个乘法器和m*p个加法器，面积功耗要求高，属于用空间换时间的方法

具体方法是，对于A矩阵（m*n）和B矩阵(n*p)，一次取矩阵A的第k列元素（$a_{i,k}$）和矩阵B的第k行元素（$b_{k,j}$）把他们两两相乘，可以得到$c_{i,j}$，再加到对应位置上去。把k从1取到n,就可以全部算完。理论上这是不是最优解？因为它把原矩阵的每个数都只取了一次，在时间上应该不能再少了。

传统方法逐行逐列相乘只用n个乘法器，但需要m*p个时钟周期完成，属于用时间换空间的方法

哪个好？不太清楚，能否混合在一起？采用矩阵分块思想？

具体的，外层采取传统，内层采取加速，最终得到的加法/乘法器数目与行分块的数目有关，显然均匀划分硬件利用率最高，设分了w块，则最后需要$m*p/w^2$个加法/乘法器。最终得到的周期数只与行分块数有关，最后需要nw^2个周期。代入数据m*p=256,n=16得到，$w=1$，2，3，4，最终应该通过实验测定。如果想要均匀分的话，$w=1$，2，4，情况并不多，优化效果有限？

这种方法确实利用了fpga的并行性，但是是否存在同一时钟大量访存的问题？

还有一种方法是通过脉动阵列实现，原理如下博客

脉动阵列学习 - 知乎

需要m*p个乘法器加法器，需要n+m+p-2个时钟周期实现，感觉不如上面的，但是这好像是比较公认的做法，应该是因为规整流水线化，易于布局布线。

或许可以问一下老师有没有推荐的方法？