矩阵乘法

以 A左乘(B=AS+E为例)

S首地址$addr_S$,E首地址$addr_E$

寄存器:

ptr_A:记录A的四个数读到第几个数

ptr_{col}:记录当前运算到S的第几列

ptr_{row}:记录当前运算到S的第几行

ptr_{addr}:当前工作地址

ptr_{addr_E}:E的地址,也是结果所在地址,与ptr_{addr}实际上应当同时变化

终止条件:$ptr_A = 4, ptr_{col} = 1344, ptr_{row} = 1344$

$MAC_B = Hash[ptr_A]$

$MAC_A[i] = MEM[ptr_{addr} + i]$

$MAC_C[i] = MEM'[ptr_{addr_E} + i]$

算出的结果对E进行覆盖

$ptr_{addr} = addr_S + 1344 * ptr_{row} + ptr_{col}$ (这个不需要实际的加,它只需要每次加4和复位至$addr_S$即可)

$ptr_A == 4 \& ptr_{col} == 1344$时,Hash进行计算,复位$ptr_A$

1. $ptr_A = (ptr_{col} == 1344)?ptr_A + 1 : ptr_A$
2. $ptr_{col} = (ptr_{col} == 1344)?0 : ptr_{col} + 4$

(不写了,再写也只是纯体力活,使用几个工作寄存器应该就可以把整个乘法过程顺下来,感觉没什么问题)

涉及到的全部矩阵乘法有

B=AS+E,B'=S'A+E',V=S'B+E'',M = C-B'S

因此,B和E,B'和E',V和E''会存在相同位置

S和E,S'和E',S'和B和E''必须存在不同位置

3个64位RAM足以满足条件

M = C-B'S下面再讨论

3个64位RAM足以满足条件

乘法以外的操作

主要应该就是上面涉及的矩阵的存储了(我对流程细节不是特别清楚,不太确定)

是否进行采样一个选择器就可以搞定

由于S和E得存在不同的RAM里但是它们还是得一次shake里出,指令上不太好搞,可能还是得状态机(其实就算是矩阵乘法,用指令也得有个状态机),或者就是把生成S'和S的拆成两个指令>这个有待商榷,反正状态机跳不开

S^T, S'存在0号RAM

E, E'存在1号RAM

E''存在2号RAM(但是这个本身只有8x8x16bit,应该可以不存了),如果不存,乘法器的C就多个选择器

M = C- B'S

这个是写的时候发现的,一个是C好像会是输入,另一个是它里面有减法,以下想到哪写到哪了

按照之前的涉及,主要涉及的问题是:

1. M的存储地址
2. 减法

存储地址上,本身好像没多大,用个寄存器就行,或者和E''一起在2号RAM,固定地址,只是这个RAM比较小

减法得考虑一下乘法器内部的设计,应该问题不大?