Homework 4: Systolic Array

信息科学技术学院 吴欣桐 2100012708

Problem 1

由T可得到P=[0 1 0 / 0 0 1]，Π=[1 1 1]。dA=[0 1 0]，dB=[1 0 0]，dC=[0 0 1]

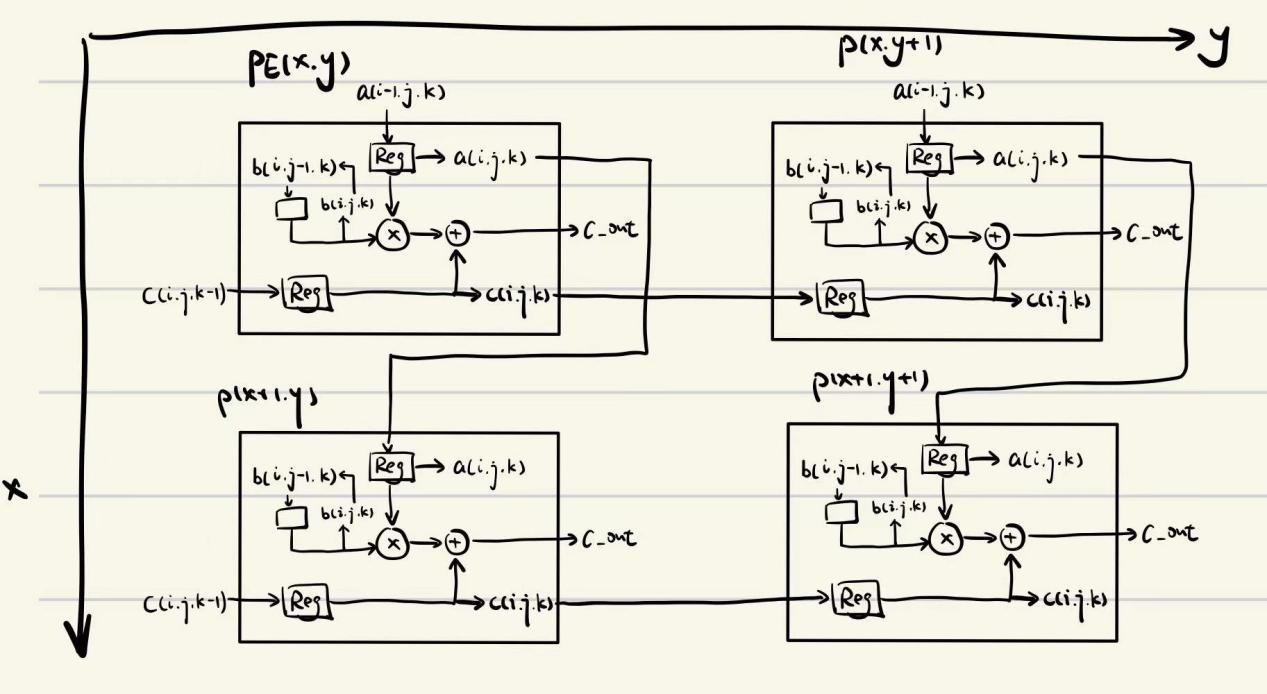
P×dA=(1,0)，Π·dA=1。则Tensor A为沿着 x 轴正方向 systolic

P×dB=(0,0)，Π·dB=1。则Tensor B为stationary，沿y轴正方向

P×dC=(0,1)，Π·dC=1。则Tensor C为沿着 y 轴正方向 systolic

且均需加register。

PE结构形如



代码设计上，对A有a(i)(j) := a(i - 1)(j)，对B有当stationaryCtrl=1时更新b(i)(j) := b(i)(j - 1)，对C有c(i)(j) := a(i)(j) \* b(i)(j) + c(i)(j - 1)。输出c\_out(i) := c(i)(3)。（此上均需考虑ij所有遍历，为0时特例不赘述）

Problem2

由T可得到P=[1 0 0 / 0 0 1]，Π=[0 1 0]。dA=[0 1 0]，dB=[1 0 0]，dC=[0 0 1]

P×dA=(0,0)，Π·dA=1。则Tensor A为stationary，沿y轴正方向

P×dB=(1,0)，Π·dB=0。则Tensor B为沿着 x 轴正方向 systolic

P×dC=(0,1)，Π·dC=0。则Tensor C为沿着 y 轴正方向 systolic

注意到仅有A需加register，B和C的更新是瞬时的，即更新第0行/列时无寄存器延迟一个周期。

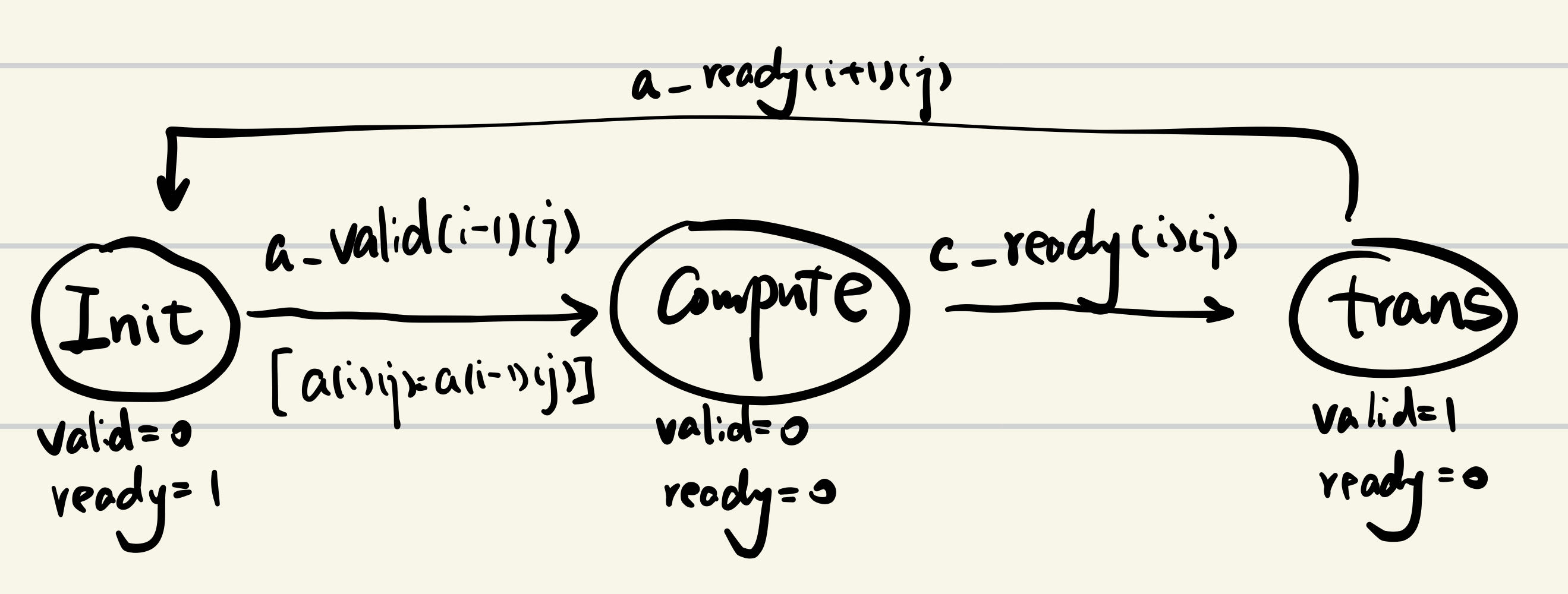
PE结构和Problem1同理，注意BC应去掉Register。

代码设计上，对A有stationaryCtrl=1时更新a(i)(j) := a(i)(j - 1)，考虑BC无寄存器的情况下，c(0)(j) := c\_in(j) + ( for (i <- 0 until 4) yield a(i)(j) \* b\_in(i)).reduce(\_ + \_)，c(i)(j) := c(i - 1)(j)。输出c\_out(i) := c(2)(i)（结合testbench考虑）。

Problem3

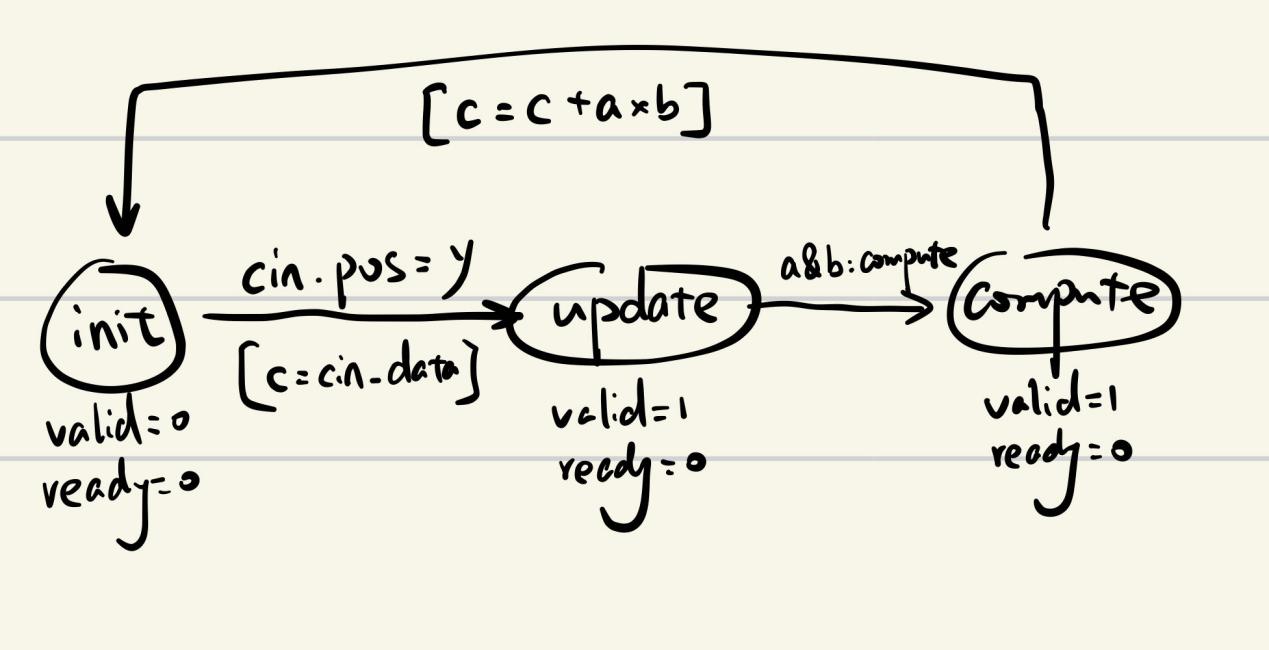
考虑所有信号用握手协议表示，配置五组寄存器a/b/c/cin/cout，分别涵盖data，ready，valid三组信号。分别将对应ready，valid，data信号和a\_in/b\_in/c\_in/c\_out相连。

考虑TensorA为 X 正方向 systolic，Tensor B为着 Y 正方向 systolic，更新状态机采用moore机，如下所示：其中在fifo常规两个状态间增加compute状态，用来更新c值。b状态机和a原理相同，仅为方向不同，不再赘述。TensorA和B的值用寄存器a和b来记录。



TensorC为Y方向stationary，采用常规FIFO实现，值用寄存器cin记录。

寄存器C采用Moore状态机，控制准备输出的结果是更新还是保存，由于涉及到cin的变化，应有更新c=cin（当pos和Y坐标相同）和更新c=c+a\*b（ab均在compute状态）两种计算。状态转移图如下所示。



Cout也为常规FIFO实现，区别为当前位的cout不仅可以由前一位经握手协议传输数据更新，也可以由寄存器C更新c=cin时更新，两条路径互不冲突。

输出有c\_out(i).valid := cout\_valid(i)(3)，c\_out(i).bits := cout(i)(3)。