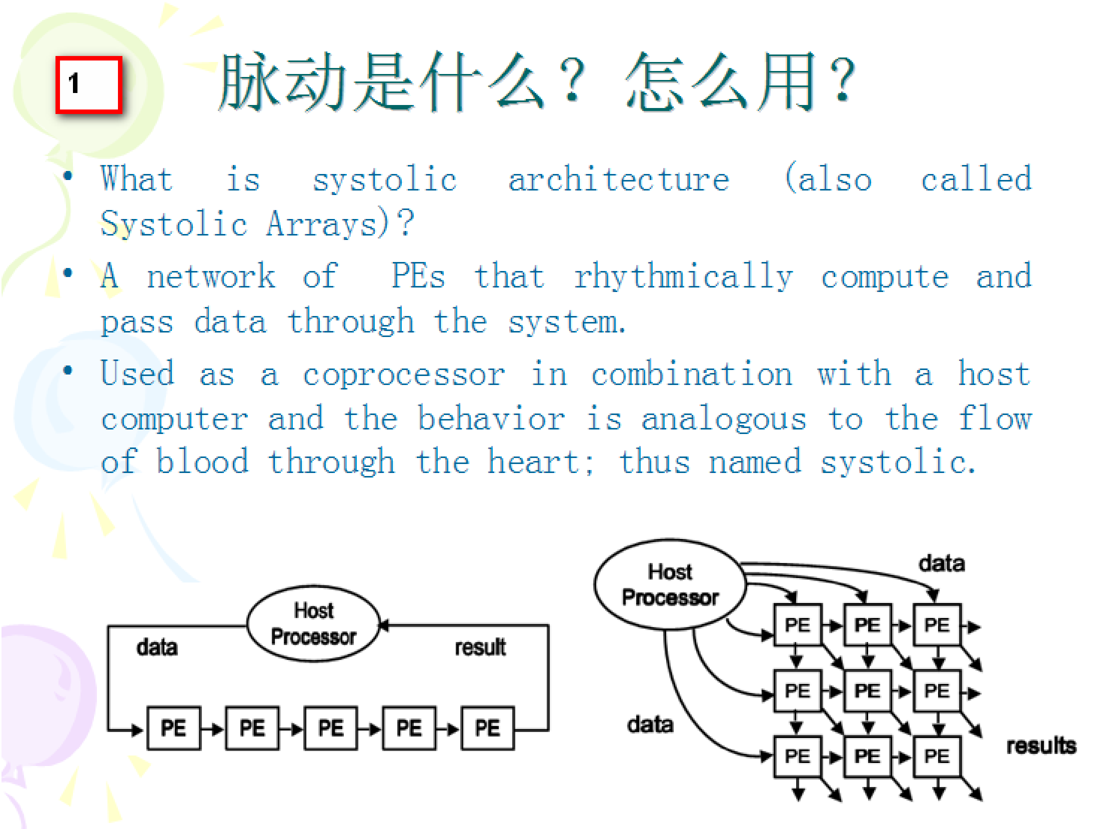
这一章，与大家一起讨论四种上古神器（包括：重定时、展开、折叠和脉动）的最后一件：脉动阵列。说起来这项技术还是咱们“龙的子孙”发明的，来自卡内基梅隆大学（现在哈佛？）的孔祥重教授。

不过在正式讲解脉动阵列之前，有必要给大家打个预防针：脉动阵列在所讨论的四件神器中应该是最费脑筋的一件，要想真正理解并掌握脉动阵列，不仅需要良好的立体几何思维，也需要超人的细心，同时还需要一点点创造性思维。这种难啃的东西，很多人会放弃，但如果你深入进去，你肯定着迷。

这里所讨论的脉动阵列只能说是入门，很多学者正努力扩展传统的脉动阵列理论，以拓宽脉动阵列的应用领域。对这方面感兴趣的同学可以 google 看文献深入学习，这里我们的目的就是讲解最简单的脉动阵列设计技术，带领大家入门，至于登堂入室那是你自己的事情咯！



脉动阵列到底是什么呢？如幻灯片 1 给出的一维脉动阵列（线形）和二维脉动阵列（矩形），它们与主处理器的关系就像是“心脏和脉络”的关系，脉动阵列不断的接收从处理器泵出的待处理数据，然后从另一边将处理后的结果 传回处理器。

较为正规的定义：多个相同的处理单元（简称 PE）按一定互联规则组成的网络，称为脉动阵列。脉动阵列可以是一维线形、二维三角形、二维矩形、二维六边形、二维二叉树型、三维长方体形等等。**脉动阵列（这种 PE 网络）的特点是：**

1. 每一个节点，也就是 PE，也称为胞元，都是相同的。
2. 每个 PE 只与其相邻的 PE 进行通信，也就是说 PE 之间的通信具有局部性，而且通信是规则的。可想而知如果通信不是局部的而且不规则，那么网络中各 PE 的连接关系将会很错乱，硬件上进行布局布线也会遇到困难。
3. 每个 PE 都有其局部的存储器，也就是 PE 的某些边带有延时，延时在硬件上对于寄存器。这说明脉动阵列数据储存具有局部性，同时这也是流水运行的必要条件。

由于脉动阵列的以上特点，造成 PE 之间的高度流水化、规则化，因此系统吞吐率非常大且易于 VLSI 的实现。流水化意味着吞吐率大，规则化则意味着版图流片成功率大。

以上所说的脉动阵列特点是一种理想的特点，工程上为了扩大脉动阵列的用途，会引入一些

弛豫，比如允许使用邻近（靠近但不是相邻）互联，使用数据广播操作，以及在系统中使用不同的胞元，尤其是边界上的胞元往往和网络内部胞元不太一样。

幻灯片 2 给出了脉动阵列的 keywords，大家可以在看完这一章，看懂这一章之后好好来品味这些 keywords 所代表的含义。

虽然还没开始脉动设计技术的讨论，但是通过前面的铺垫，大家应该知道这么一点：脉动阵列是高度流水化和规则化的多处理器网络（注意，处理器/胞元/PE 为同一个东西，均指脉动阵列的一个处理单元）。

既然脉动阵列是高度规则的，那么脉动阵列所完成的功能是不是也应该是规则的呢？

这是脉动入门的第一道坎，一定要记住：不是任意的算法都可以用脉动阵列来实现，只有规则的迭代算法，才能用投影技术设计出脉动结构。问题又来了，

怎么判断一个迭代算法是不是规则的？FIR 是规则迭代吗？矩阵乘法是不是规则迭代？其他等等……

判断一个迭代算法是否规则，首先画出该算法的依赖图（DG），关于 DG 是什么，在 第一章、敲门砖——入门的准备 的内容有讲到，这里我们假设迭代算法的 DG 是已知，只讨论

如何根据规则 DG 设计出脉动结构。比如三阶 FIR 滤波器和 2x2 矩阵乘法的 DG 如下图 1 和图 2 所示，

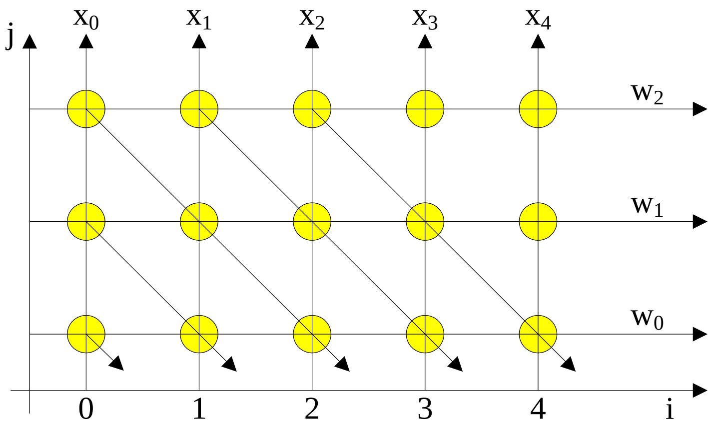


图 1　　三阶 FIR 依赖图，y(n)=w0*x(n)+w1*x(n-1)+w2*x(n-2)

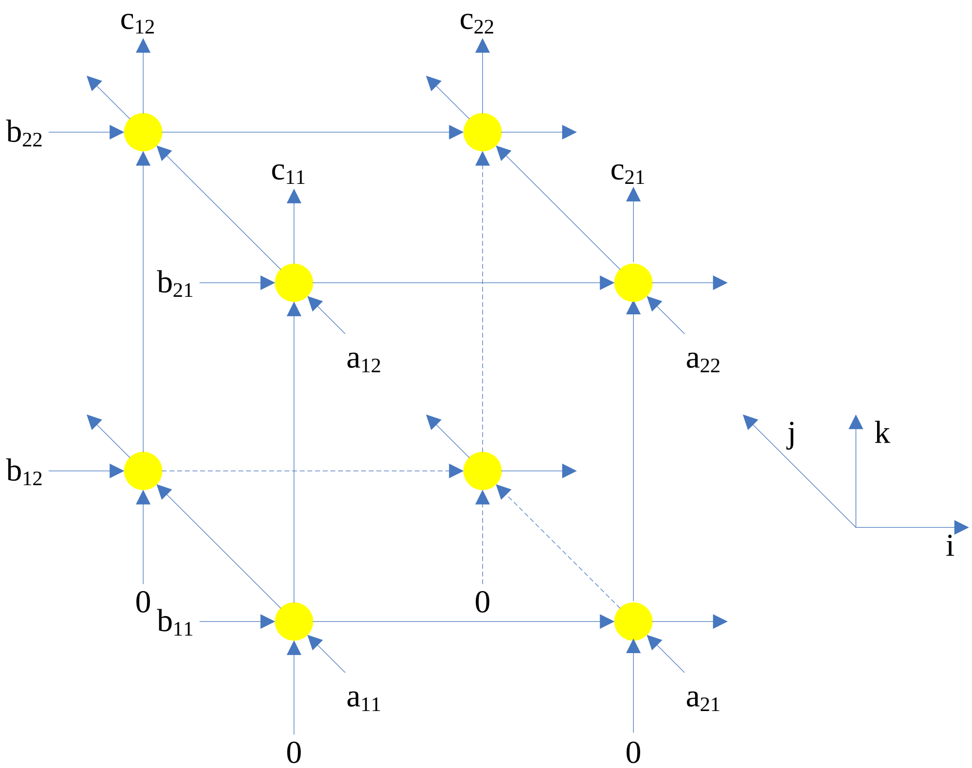


图 2　　2x2 矩阵乘法 C=AxB

请大家根据第一章中对 DG 的解说来验证这两个 DG，并总结出一些根据规则迭代公式画出 DG 的做法；这两个 DG 也正是课本上例子。

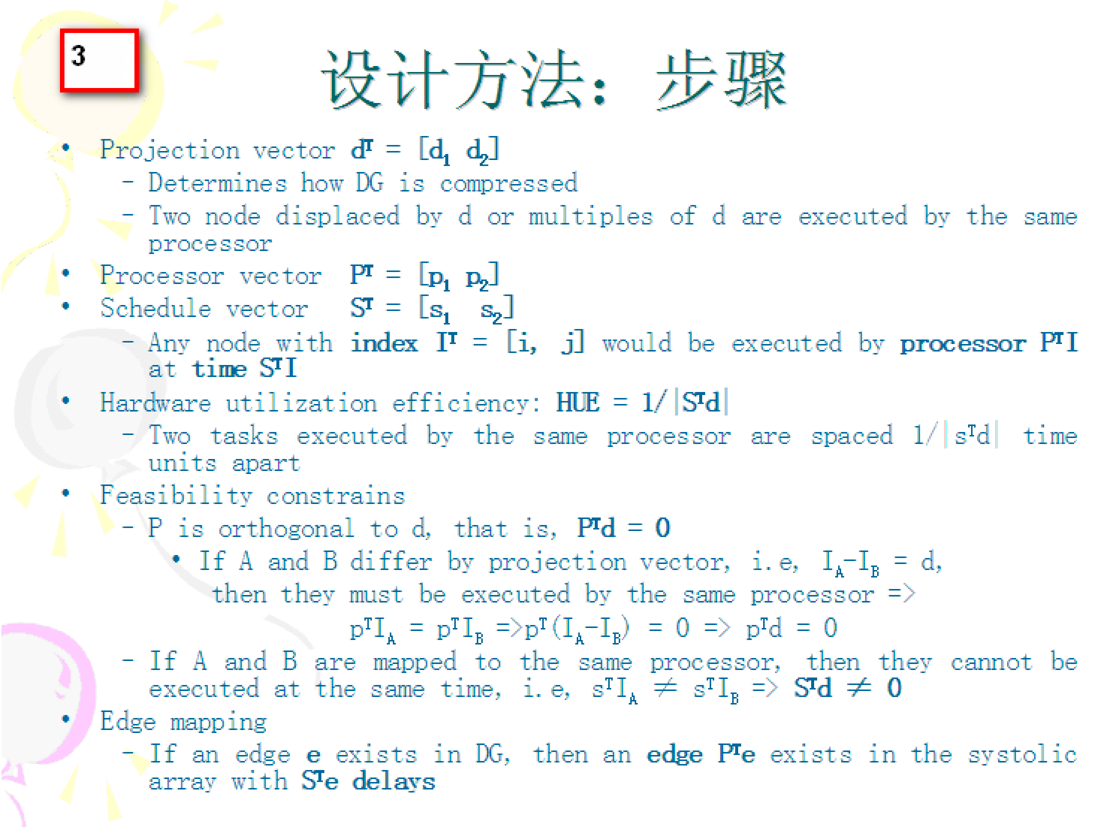
规则 DG 的判据：

如果依赖图的任一节点沿某个方向的边存在，则称依赖图是规则的；通俗的说，依赖图的所有节点具有相同形式的边。

例如图 1 的每个节点都可以看成具有三条边，一条从左向右$[1,0]^T$的权值边，一条从下到上$[0,1]^T$的输入边，一条斜向右下角$[1,-1]^T$的输出边。图 2 的每个节点也有三条边，分别是$[1,0,0]^T$方向的输入 b，$[0,1,1]^T$方向的输入 a 和 $[0,0,1]^T$方向的输出“c”。说起来，这个判据也不是绝对的严格，大家体会体会吧，也许等你大致弄明白脉动的设计方法之后，你会理解现在所说的这些话。这里值得注意的是，我们用$[0,0,1]^T$之类的符号表示方向，比如在图二中，清楚的标出了坐标系 i-j-k，那么$[0,0,1]^T$就表示向上的一个方向，又比如在图一中是以 i-j 构成坐标系，那么$[1,-1]^T$就表示斜向右下角的方向。

幻灯片 3 给出脉动阵列的设计步骤，大家可以“先死记硬背”着。接下来的内容将以 3 阶 FIR 和 2x2 矩阵相乘为例，详细讨论如何来导出课本上所列出的各个设计结果。

题外话：我自己在学习这一章的时候，看到这里一直是晕晕乎乎的状态，根本不解脉动阵列是什么东西。但是在例子的学习中，突然开窍了，也明白了所有前面这些内容的意思，所以在以下例子的讨论中，大家一定要咬紧牙关，不论你是在和我一起学习，还是你在自学，只要你开动脑筋开足马力去研究这些课本上这些例子，肯定能开窍。



例一、3 阶 FIR 滤波器的脉动阵列设计，DG 如图 1 所示。

脉动阵列设计的方法有很多，比如代数法、参数法、变换法和 投影法等等。我们所讨论是投影法，也是最直观的一种方法。在立体几何中，所谓的投影是什么？

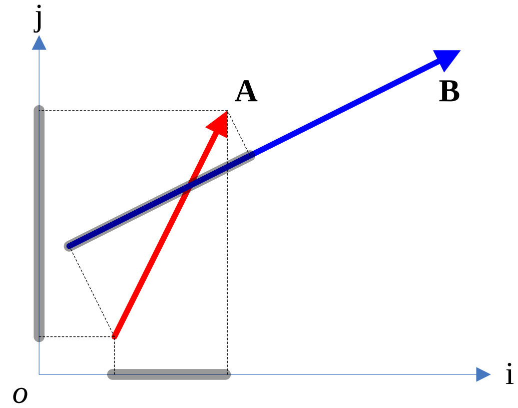


图 3　　矢量 A 在坐标轴上的投影

如图 3 示，矢量 A 在 i 轴和 j 轴上的投影，以及矢量 A 在矢量 B 方向上的投影，矢量在某个方向上的投影对应的数学运算就是内积。

如图 1 的 3 阶 FIR DG，表示了所有（无限次）迭代的节点依赖关系，为了在硬件上进行实现，必须将 DG 映射到实际电路，也就是说必须将 DG 中的节点和边分别映射为具体的硬件处理单元和互联关系（或者是带延时的互联关系）。在我们要讨论的脉动设计技术中，这种映射是通过投影来实现的，具体而言，如图 4 示

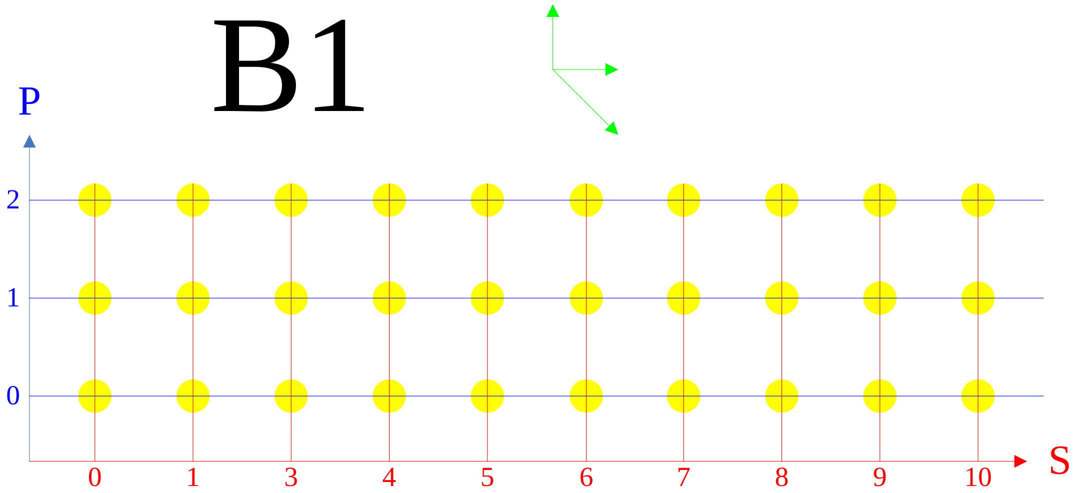


图 4　　B1 脉动阵列映射图（输入广播，结果移动，权重保持）

我们把 DG 图所在的 i-j 空间投影到一维脉动阵列空间。一维脉动阵列是“这么样”的一个 2 维空间，一个维度是 PE 空间，也就是说 PE 是线形的，另一个维度是时间；同理二维脉动阵列是一个 3 维空间，其中两个维度形成 PE 空间，显然 PE 就是平面网络，可能是矩形也可能是三角形或六边形等，另外一个维度是时间。如图 4，蓝色坐标轴 P 是 PE 轴，红色坐标轴 S 是时间轴，图中的水平蓝线和竖直红线清楚的显示了节点（黄色）是如何投影到 PE 轴和时间轴的，这种投影的物理意义是：所有在同一条竖直红线上的节点在同一个周期被调度执行，比如从左边起的第一列三个节点投影到 S 轴的 0 位置，那么这三个节点将在周期 0 被调度执行，同理第二列的三个节点将在周期 1 调度执行，也就是说一个节点投影到 S 轴的哪个位置，就表示在那个周期被调度执行。现在我们知道了每个节点调度时刻是怎么

来计算的，那么节点要调度到那个 PE 上执行呢？在这个例子中使用了一维脉动阵列，线形的 PE 网络，类似像时间轴投影的过程，所有节点向 P 轴投影，投影的位置就是节点运行所在的 PE 位置，比如从下到上三条水平蓝线上的节点分别被投影到 PE0、PE1 和 PE2 三个硬件处理器上运行。

说到这，大家也许有些明白了。设计脉动结构，其实就是对 DG 中的各个节点进行调度，也就是说 xxx 节点安排在 xxx 周期调度到 xxx 处理器上运行。其实本质上就是节点调度的问题，只是这种调度不是任意的，要符合一定的规则，这样才能保证所得脉动阵列功能是正确的。**仔细思考下面这个问题：**约定一个 PE 在一个周期内只能执行一个节点的计算任务，也就是说，不能在同一个周期内将多于一个的节点映射到同一个 PE。从数学意义上说，对一维脉动阵列，时间轴不能和处理器轴平行；对二维脉动阵列，时间轴不能和处理器平面平行。

开动脑筋好好想像一下，比如一维脉动阵列中，如果时间轴和处理器轴平行，那么一串被调度到同一个周期的节点，也将同时被安排到同一个 PE 来执行，这和我们前面的约定“一个 PE 一个周期只执行一个节点的计算任务”矛盾；同理，推广到二维脉动阵列，如果时间轴和处理器平面平行，又将如何呢？你来说说？

课本上将上述所说的情况定为“可行性限制条件”，也就是说，我们在构造脉动空间时，必须保证时间轴不能和处理器轴或处理器平面平行。这一点的物理意义非常明显，大家多思考一下即可明白。

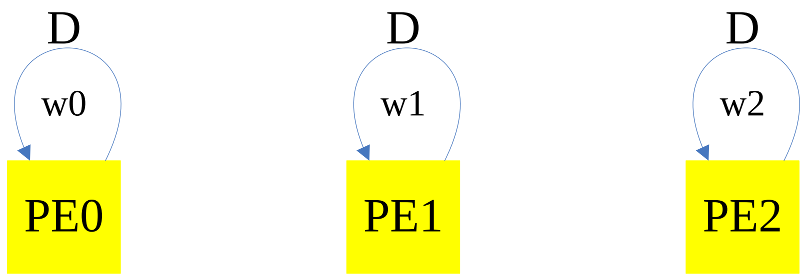
回到对图 4 的脉动设计，我们构造了这样一个脉动空间，它的时间轴与 i 轴平行，它的处理器轴与 j 轴平行，i-j 指的是 DG 空间的坐标轴，而 S-P 指的是脉动空间的坐标轴，在图 4 的例子中，构造的脉动空间恰恰好与 DG 空间“重合”。注意这只是个偶然，因为还可以构造很多其他形式的脉动空间，千万不要以为 DG 空间就是脉动空间，不要两者傻傻分不清楚。

脉动硬件电路构造：

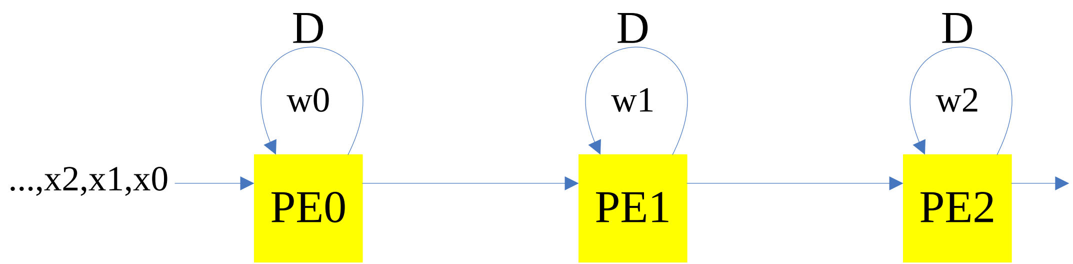
1. 从图 4 的处理器轴，可以看出所需的线形脉动网络包含 3 个 PE，0 周期左边起第一列节点调度到这三个 PE 执行，1 周期轮到第二列节点，2 周期是第三列节点，这样一直延续下去，在硬件上可以先画出三个 PE 单元，如下图

PE0　　PE1　　PE2

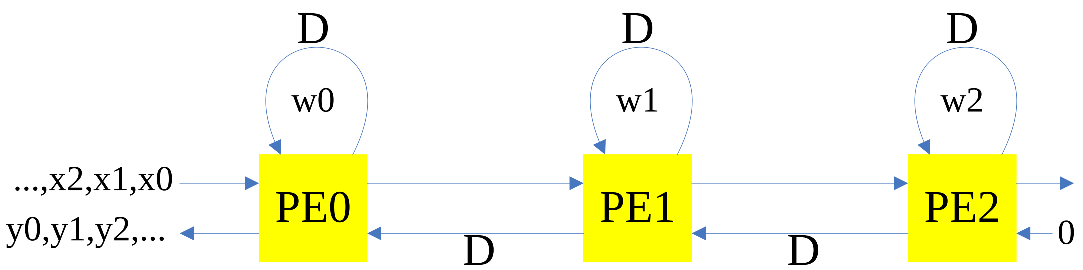
2. 接下来，需要把 DG 中节点的边映射为各个 PE 之间的互联关系。同时看图 1 的 DG 和图 4 的脉动空间映射，对于$[1,0]^T$方向的权值 w 边（水平向右），权值边连接相邻的两个水平节点，将该边投影到 S 上，跨越一个周期，将改变投影到 P 上，位于同一 PE，这就表示，权值边在硬件上是同一个 PE 上延时一个周期的连线，如下图示



再来看$[0,1]^T$方向的输入 x 边，投影到 S 没有跨度，投影到 P，是从低序号 PE 到高序号 PE 的边，反映到硬件上就是

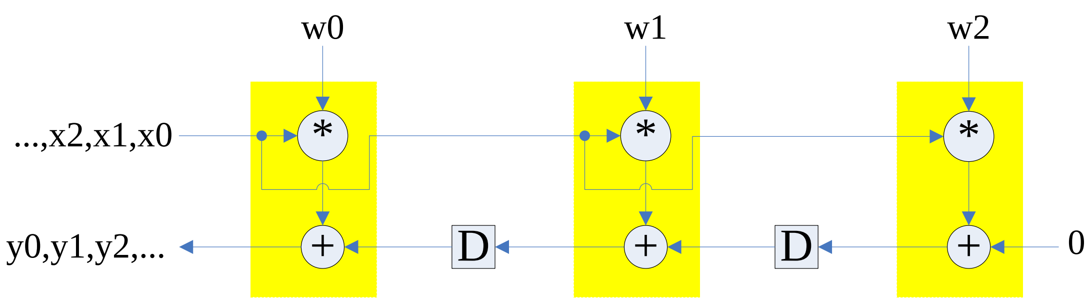


最后，在看$[1,-1]^T$方向的输出 y 边，投影到 S，跨越一个周期，投影到 P，从高序号 PE 到相邻的低序号 PE，反映到硬件上就是



至此，就完成整个 DG 到脉动空间的映射过程。建议大家牢记一点，时刻在脑海中想像其硬件的图景，清楚每一步自己都在干些什么，每一步是在硬件上又意味着什么，不要仅仅把上述的内容看出生硬的步骤!!!

3. 得出脉动结构图，结合 DG 中节点的具体内容，即可画出最终电路。图 1 节点的内容是一个乘法加法的级联单元，最终电路图如下示，其中虚线所圈为节点的具体内容，



也许大家发现，这里给出的最终电路和课本 P142 图 7-4 的电路不太一样，其实它们是同一个电路，大家在构造最终电路时，结合实际情况进行一些合理的修改是没问题的。

对于图 4 的映射过程，我们是通过观察直接构造出脉动阵列，但是对于更为复杂的映射有

时会显得力不从心。下面使用一个不同于 B1 映射的脉动空间，但这次将用严格的数学形式来描述整个过程；用数学形式来描述映射过程是非常强大的，在稍后矩阵乘法的例子中将看到，我们很难在画出形象的映射图，而只能靠“数字”来指导如何画出脉动结构。

观察图 5 的脉动空间，这里所选择的处理器轴 P 和图 4 的不一样，所形成的脉动空间 S-P 也不同于图 4。

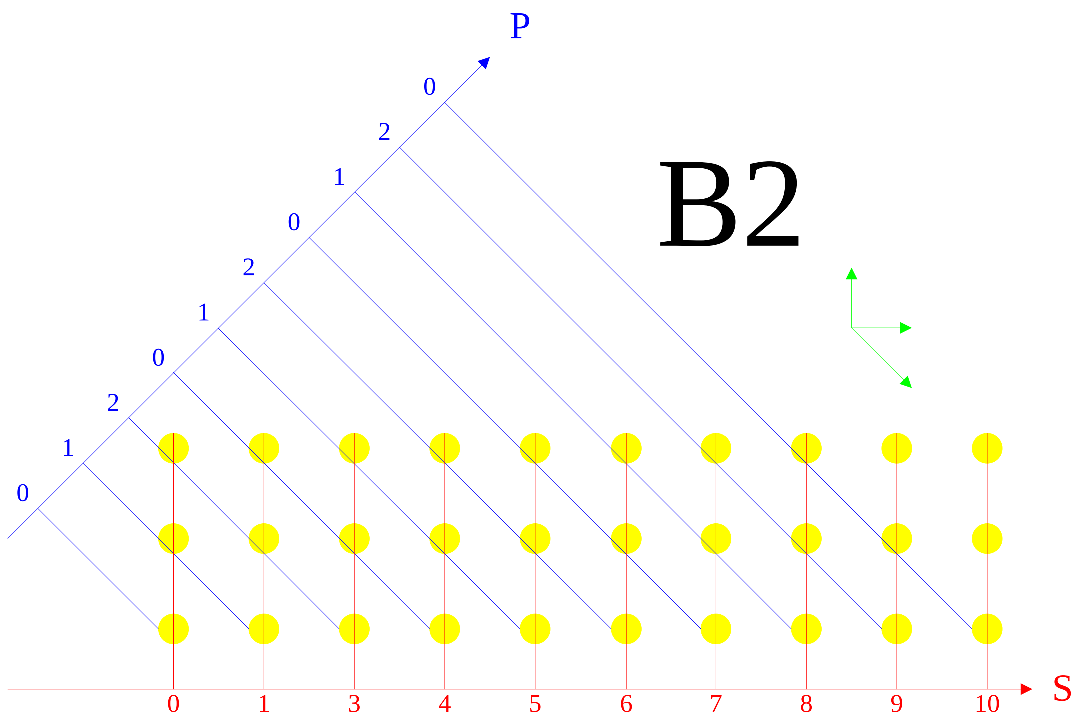


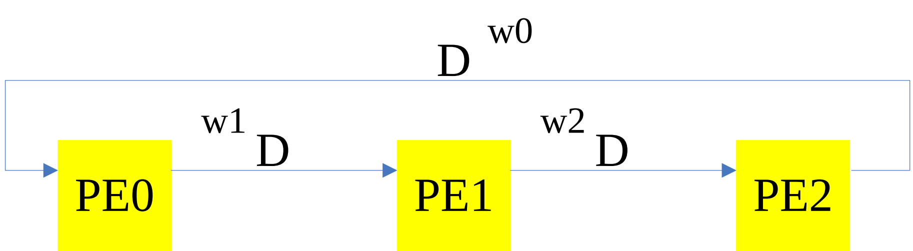
图 5　　B2 脉动阵列映射图（输入广播，权重移动，结果保持）

“严格的”脉动硬件电路构造：

1. 首先要用矢量来表示脉动空间的两个坐标轴 P 和 S，这里令 $S=\begin{bmatrix}1\\0\end{bmatrix}$，$P=\begin{bmatrix}1\\1\end{bmatrix}$，显然 P 和 S 是不平行的（数学上两个非零矢量是否平行，则它们的叉乘为 0，即 $S\times P=0$），满足“可行性限制条件”。任一个节点 $\begin{bmatrix}i\\j\end{bmatrix}$ 在 P 轴和 S 轴上的投影，在数学上就是点乘，即节点 $\begin{bmatrix}i\\j\end{bmatrix}$ 在 $S\cdot\begin{bmatrix}i\\j\end{bmatrix}$ 周期被调度到 $P\cdot\begin{bmatrix}i\\j\end{bmatrix}$ 处理器执行。例如图 5 中左下角第一个节点，坐标为 $\begin{bmatrix}0\\0\end{bmatrix}$，将在 $S\cdot\begin{bmatrix}i\\j\end{bmatrix}=\begin{bmatrix}1\\0\end{bmatrix}\cdot\begin{bmatrix}0\\0\end{bmatrix}=1\times0+0\times0=0$ 周期被调度到 $P\cdot\begin{bmatrix}i\\j\end{bmatrix}=\begin{bmatrix}1\\1\end{bmatrix}\cdot\begin{bmatrix}0\\0\end{bmatrix}=1\times0+1\times0=0$ 处理器执行；又比如左上角第一个节点，坐标为

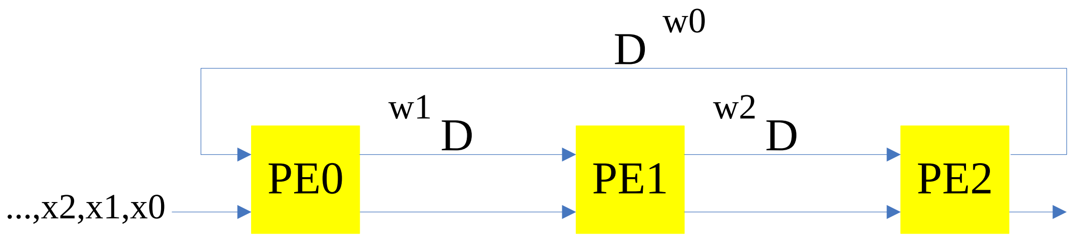
$\begin{bmatrix}0\\2\end{bmatrix}$，将在 $S\cdot\begin{bmatrix}i\\j\end{bmatrix}=\begin{bmatrix}1\\0\end{bmatrix}\cdot\begin{bmatrix}0\\2\end{bmatrix}=1\times0+0\times2=0$ 周期被调度到 $P\cdot\begin{bmatrix}i\\j\end{bmatrix}=\begin{bmatrix}1\\1\end{bmatrix}\cdot\begin{bmatrix}0\\2\end{bmatrix}=1\times0+1\times2=2$ 处理器执行，从图 5 中你显然可以看出这一点。细心的读者也许会发现图 5 中 P 轴上的坐标并不是逐渐递增的，而是 012012012…. 不断循环，为什么呢？从数学上说当节点坐标 $\begin{bmatrix}i\\j\end{bmatrix}$ 中的数字足够大，$P\cdot\begin{bmatrix}i\\j\end{bmatrix}$ 也将比 2 大才是。。。。。的确，随着节点坐标的增大，$P\cdot\begin{bmatrix}i\\j\end{bmatrix}$ 也会增大而超出 2，但是，但是，但是脉动阵列的设计就是那么具有技巧性。观察图 5，可以发现同一周期最多只有 3 个节点被映射到 P 轴的处理器执行，也就是说只需 3 个处理器便可以保证构造出功能正确的脉动阵列，而不是无限个处理器，如果仅仅看 $P\cdot\begin{bmatrix}i\\j\end{bmatrix}$ 所得出的数字，会得出必须有无限个处理器才能构造出脉动阵列来。对于图 5 的情况，可以循环利用 3 个处理器即可实现脉动阵列。类似 B1 的设计，这里先画出三个 PE 单元，如下图

2. 在 DG 中存在三条边，分别是$[1,0]^T$的权值 w 边、$[0,1]^T$的输入 x 边以及$[1,-1]^T$的输出 y 边，如图 5 的绿线所示。注意这些矢量不仅表示了这些边的方向，也表示了这些边的长度，不要以为$[1,0]^T$和$[2,0]^T$代表相同的边，它们只是方向相同而长度不同。将$[1,0]^T$的权值 w 边分别投影到 S 轴和 P 轴，有 $S\cdot\begin{bmatrix}1\\0\end{bmatrix}=\begin{bmatrix}1\\0\end{bmatrix}\cdot\begin{bmatrix}1\\0\end{bmatrix}=1$ 和 $P\cdot\begin{bmatrix}1\\0\end{bmatrix}=\begin{bmatrix}1\\1\end{bmatrix}\cdot\begin{bmatrix}1\\0\end{bmatrix}=1$，也就是说权值脉动阵列中是有低序号 PE 流向高序号 PE 且跨越一个周期。注意根据前面我们对 P 轴所作的修改，3 号 PE 其实循环变为 0 号 PE，注意这种循环关系。到此，脉动结构图如下

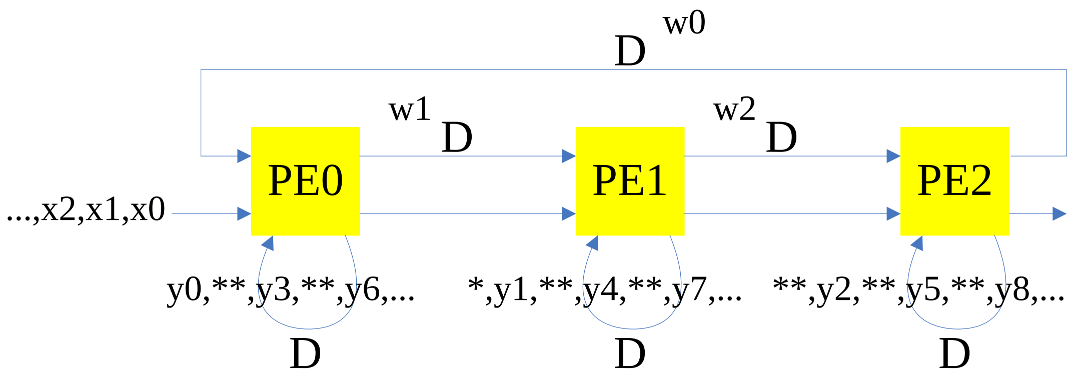


结合图 5 好好想像一下实际电路中发生的情况吧，你会发现的确这些权值就是在各个 PE 之间这样循环流动的。接下来是$[0,1]^T$ 的输入 x 边，分别投影到 S 轴和 P 轴，有 $S\cdot\begin{bmatrix}0\\1\end{bmatrix}=\begin{bmatrix}1\\0\end{bmatrix}\cdot\begin{bmatrix}0\\1\end{bmatrix}=0$ 和 $P\cdot\begin{bmatrix}0\\1\end{bmatrix}=\begin{bmatrix}1\\1\end{bmatrix}\cdot\begin{bmatrix}0\\1\end{bmatrix}=1$，这就意味着输入 x 也是从低序号 PE 流

向高序号 PE，但没有时间上的延迟，也就是数据广播结构。

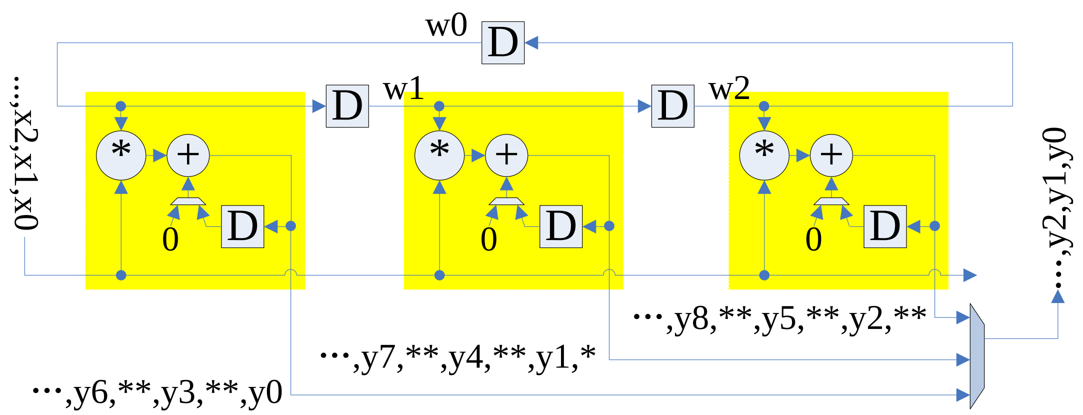


最后是$[1,-1]^T$的输出 y 边，分别向 S 轴和 P 轴，有 $S\cdot\begin{bmatrix}1\\-1\end{bmatrix}=\begin{bmatrix}1\\0\end{bmatrix}\cdot\begin{bmatrix}1\\-1\end{bmatrix}=1$ 和 $P\cdot\begin{bmatrix}1\\-1\end{bmatrix}=\begin{bmatrix}1\\1\end{bmatrix}\cdot\begin{bmatrix}1\\-1\end{bmatrix}=0$，这就是说结果 y 在同一个 PE 上循环且延时一个周期，即



在这个脉动结构中，结果的输出是比较诡异的，但是如果你看懂了图 5，也就不奇怪了。从图 5 中可以看出 0 周期在 PE0 输出 y0，1 周期在 PE1 输出 y1，2 周期在 PE2 输出 y2，3 周期在 PE0 输出 y3，等等，表现在硬件上就是上图所画的形式，实际实现通过多路选择器在特定周期选同某个 PE 上存储的结果作为最终的 y 输出，而且该 PE 结果输出之后，要对其寄存器（循环边上的 D）进行置零初始化。

3. 最终电路如下图是，其中还需要添加一些控制电路，用于产生选路器的选择信号。大家可以自己动手试试。



结合数学公式来推导脉动结构，可以使得我们能处理更为复杂的问题而不受大脑想像能力的制约。接下来，将把课本上的所有 3 阶 FIR 滤波器的脉动结构过一遍，但是不会像上面的

那么详细，这也正是大家练手的机会，试着把最终电路画出了，并弄清楚数据（包括权值，输入和输出）是怎么在硬件电路中流动的。

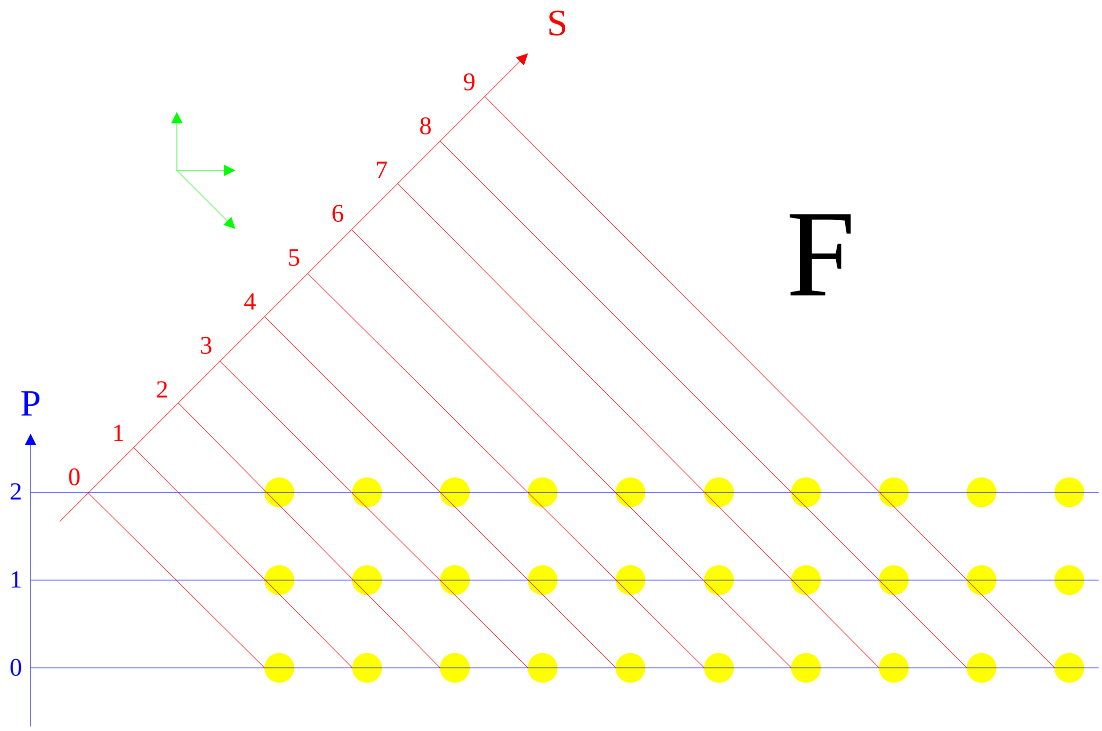
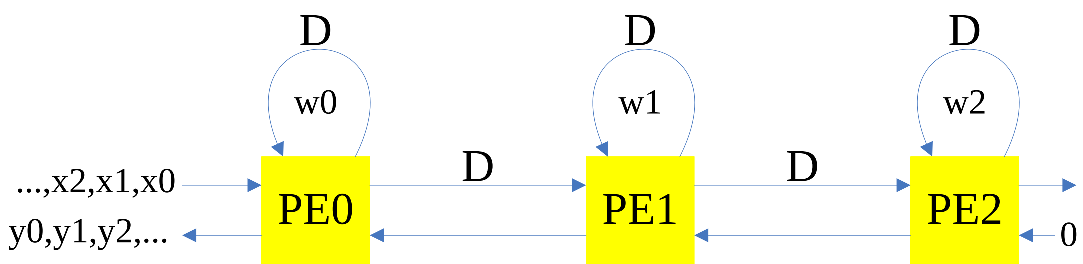


图 6　　F 脉动阵列映射图（结果扇入，输入移动，权重保持）

如图 6，选择 $P=\begin{bmatrix}0\\1\end{bmatrix}$ 及 $S=\begin{bmatrix}1\\1\end{bmatrix}$，显然 P 和 S 不平行，满足"可行性限制条件"。根据节点对 P 轴的映射情况，只需设置 3 个 PE 节点。按课本上的形式，列出三条边的映射结果表，

边	$P^T e$	$S^T e$
权值 $[1,0]^T$	0	1
输入 $[0,1]^T$	1	1
输出 $[1,-1]^T$	-1	0

脉动结构为



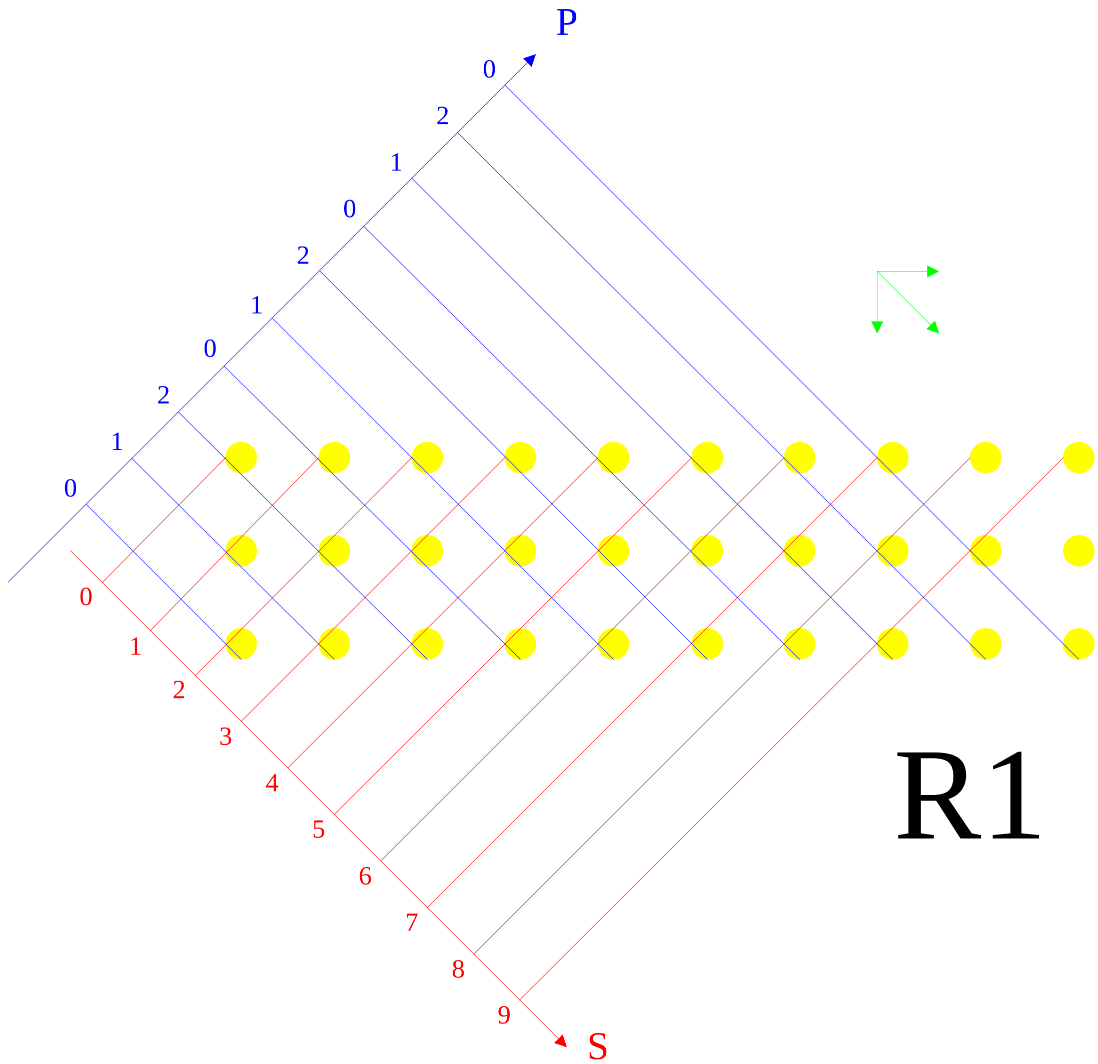
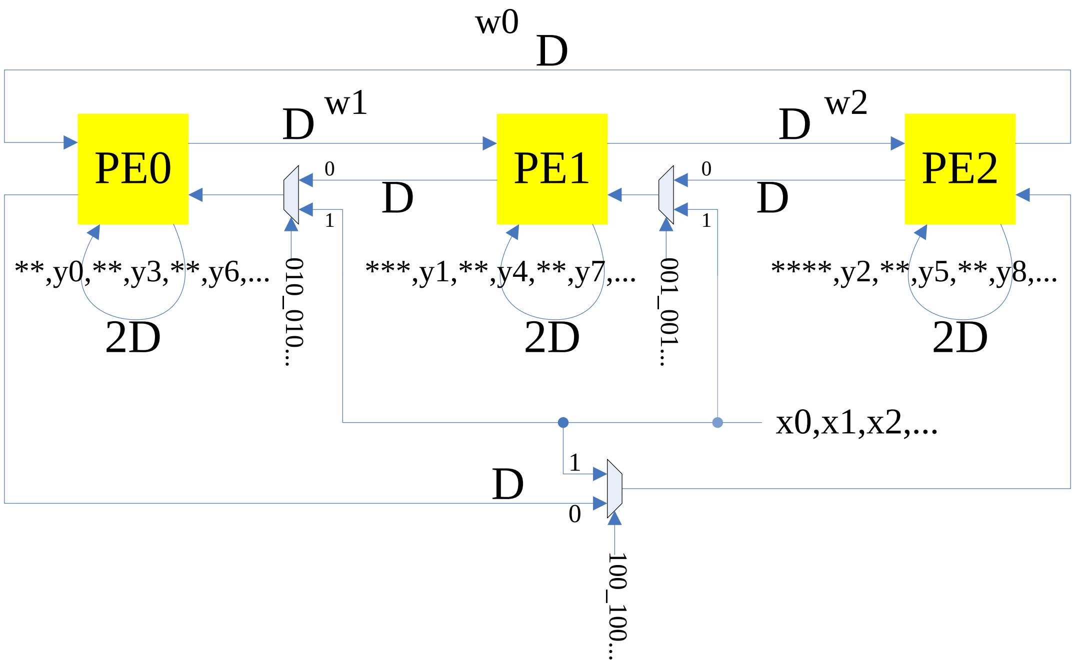


图 7　　R1 脉动阵列映射图（结果保持，输入和权重反向移动）

如图 7，选择 $P=\begin{bmatrix}1\\1\end{bmatrix}$ 及 $S=\begin{bmatrix}1\\-1\end{bmatrix}$，显然 P 和 S 不平行，满足“可行性限制条件”。思考题：你能看出 R1 结构处理器轴应该如何处理吗？仔细观察 R1 结构，同一周期最多也只有 3 个节点投影到 P 轴，也就是说只需 3 个 PE 即可，就和前面讲到的 B2 结构一样。在 R1 结构中，直接进行边映射会出现跨越“负周期”的情况，这在硬件上是不可实现的，好在图 1 所示的 DG 的输入边是可以反转的，即从上指向下；其实权值边和结果边也是可以反转的。DG 中的边是否可以反转，必须结合实际情况来分析，如果你看懂了这个 FIR 的 DG，肯定也就明白为什么说该 DG 的所有边都可以反转。有时我们要对某些边进行反转才能继续脉动阵列的构造。在后面的一些结构中，还会反转其他的边。在 R1 中只需反转输入边，三条边用绿色标注在图 7 中。边的映射结果表如下

边	P^Te	S^Te
权值 $[1,0]^T$	1	1
输入 $[0,-1]^T$	-1	1
输出 $[1,-1]^T$	0	2

脉动结构为



注意，y0 是在第 2 周期从 PE0 上输出，接着是第 3 周期 y1 从 PE1 上输出，等等类推。输入稍微复杂一些，主要是因为输入的入口是随着时间循环的，PE2/PE0/PE1，如此在输入的电路上就需加上选路器和控制逻辑。

奇怪！

课本上的电路与我自己构造出来的不太一样？但我想大概是对 P 轴上 PE 的安排有关。哪位同学有兴趣深入研究，可以把自己的想法发表出来。同时，也要声明一下，以上讨论仅仅在个人的脑子中验证过，并没在仿真器里运行，所以不保证 100% 正确，大家要留心，不能盲从。如果发现我的错误，也欢迎大家指正，谢谢！

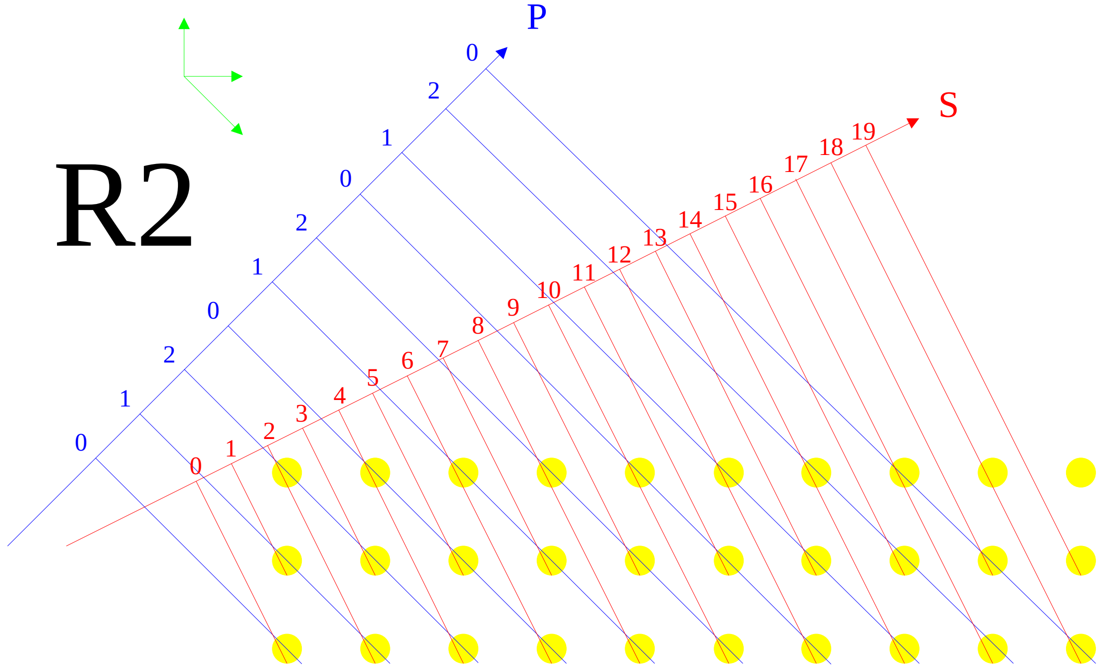
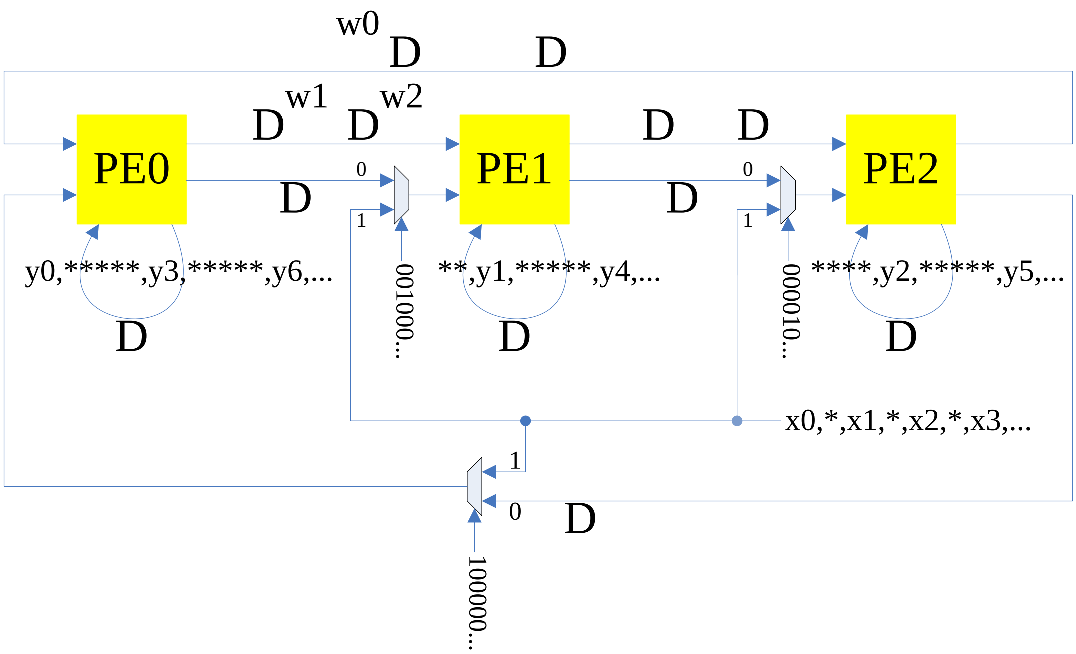


图 8　　R2 脉动阵列映射图（结果保持，输入和权重同方向但不同速度移动）

如图 8，选择 $P=\begin{bmatrix}1\\1\end{bmatrix}$及 $S=\begin{bmatrix}2\\1\end{bmatrix}$，显然 P 和 S 不平行，满足“可行性限制条件”。仔细观察 R2 结构，同一周期最多也只有 2 个节点投影到 P 轴，也就是说只需 2 个 PE 即可，但是，我们仍然在 P 轴上设置 3 个 PE，稍后再讨论设置 2 个 PE 的情况。边的映射结果表如下

边	P^Te	S^Te
权值 $[1,0]^T$	1	2
输入 $[0,1]^T$	1	1
输出 $[1,-1]^T$	0	1

脉动结构为



其实到这大家也能看出，在脉动设计中选择 P 轴和 S 轴是非常关键的，如果选择不当构造出来的脉动阵列比较复杂，而且硬件利用率也不高。例如 R2 结构，结果隔一个周期出一个，相比于前面的结构一个周期一个结果，吞吐率就下降了一半，更惨的是 PE 节点利用率也不高，有些 PE 在某些周期内是闲着的，不参与有意义的计算。

接下来，我们做一个**进阶题！** 如图 9，也是 R2 结构，但是为了提高 PE 的利用率，在 P 轴上只设置 2 个 PE。之所以可以少设置 1 个 PE，是因为同一周期最多只会有 2 个节点投影到 P 轴。注意，别以为 PE 个数可以随便设置，存在一个理论最少，这个例子中最少需要 2 个 PE。

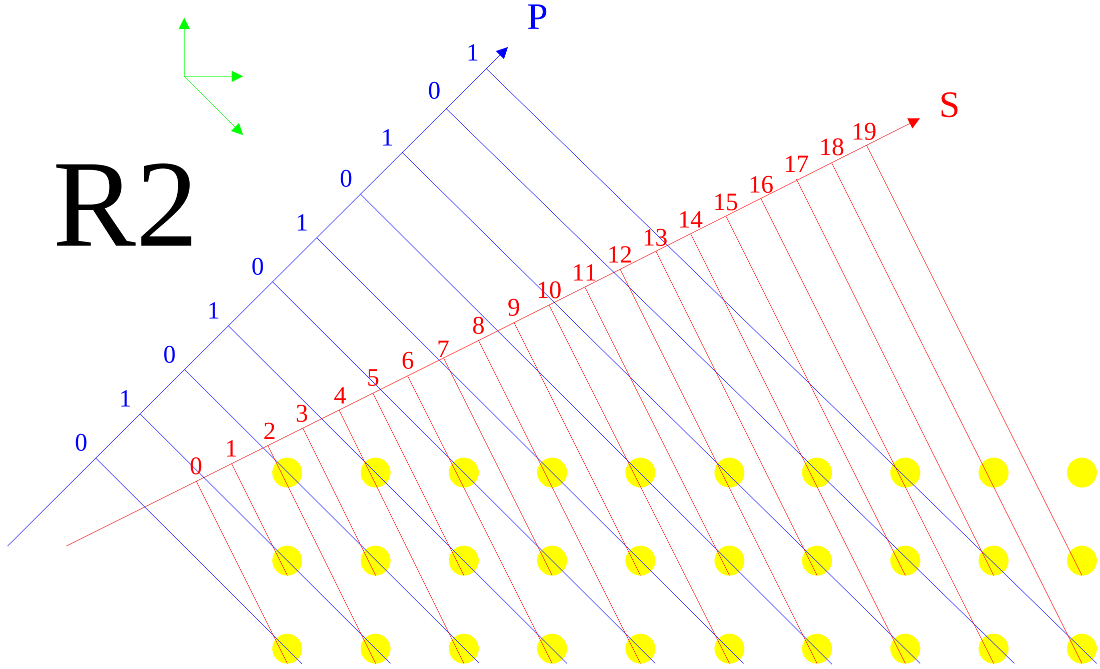
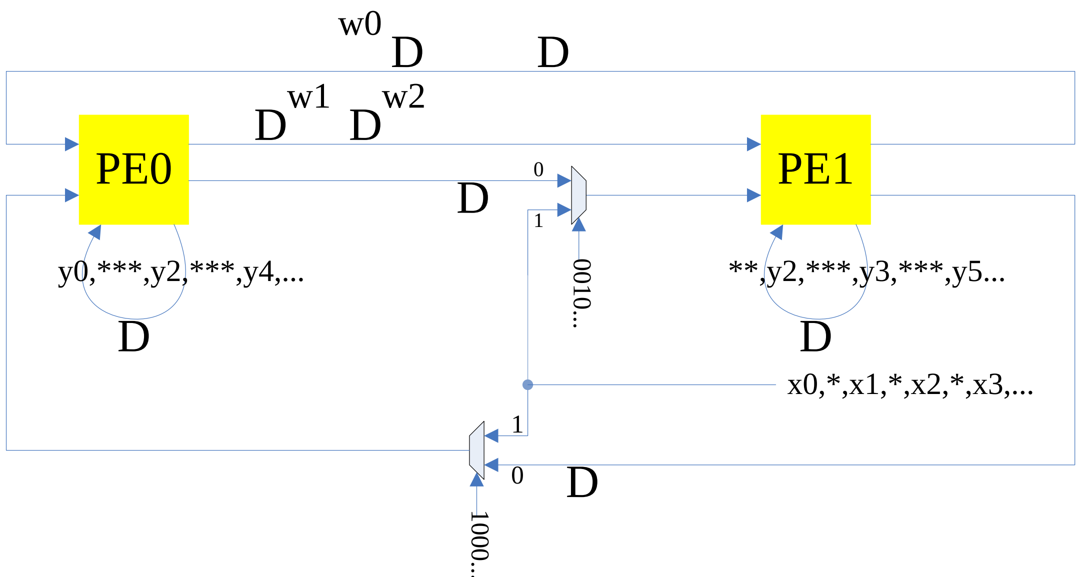


图 9 R2 脉动阵列映射，使用 2 个 PE

边映射的结果同正规 R2 的结果

边	$P^{T}e$	$S^{T}e$
权值 $[1,0]^T$	1	2
输入 $[0,1]^T$	1	1
输出 $[1,-1]^T$	0	1

那么脉动阵列是什么样的呢？你能画出来吗？



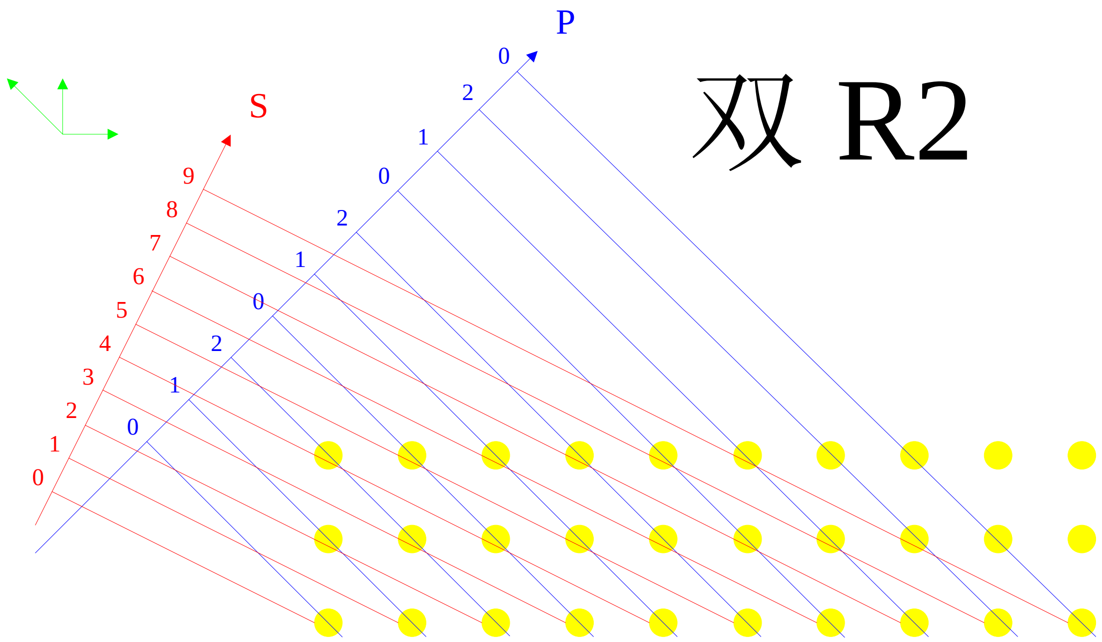
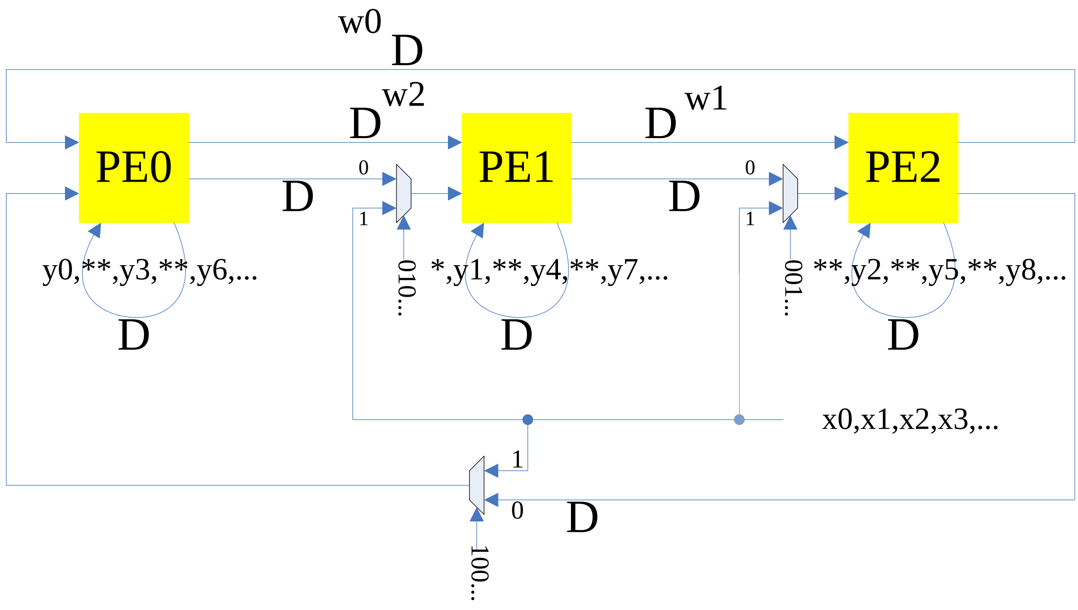


图 10　　　双 R2 脉动阵列映射图（结果保持，输入和权重同方向但不同速度移动）

如图 10，选择 $P=\begin{bmatrix}1\\1\end{bmatrix}$ 及 $S=\begin{bmatrix}1\\2\end{bmatrix}$，显然 P 和 S 不平行，满足“可行性限制条件”。对于双 R2，需要反转输出边，三条边的情况如图中绿色带箭头线所示。观察 R2 可知需要设置 3 个 PE。边的映射结果表如下

边	P^Te	S^Te
权值 $[1,0]^T$	1	1
输入 $[0,1]^T$	1	2
输出 $[-1,1]^T$	0	1

脉动结构为



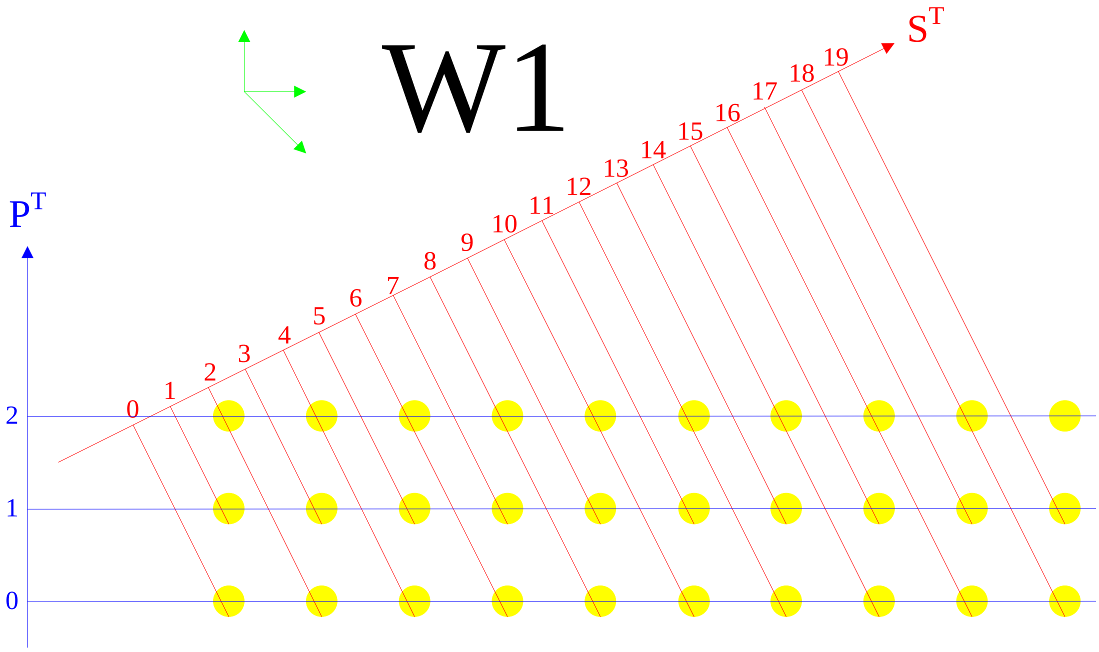
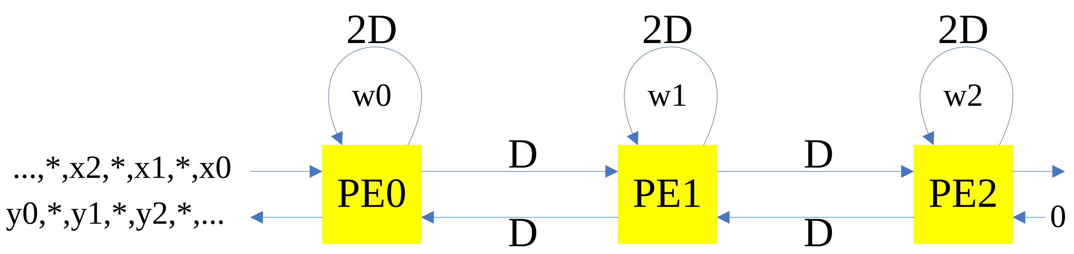


图 11　W1 脉动阵列映射图（权重保持，输入和结果反向移动）

如图 11，选择 $P=\begin{bmatrix}0\\1\end{bmatrix}$ 及 $S=\begin{bmatrix}2\\1\end{bmatrix}$，显然 P 和 S 不平行，满足“可行性限制条件”。显然 W1 可设置 3 个 PE，也可以设置 2 个 PE，设置 2 个 PE 的情况作为练习，大家自己动手试试。边的映射结果表如下

边	P^Te	S^Te
权值 $[1,0]^T$	0	2
输入 $[0,1]^T$	1	1
输出 $[1,-1]^T$	-1	1

脉动结构为



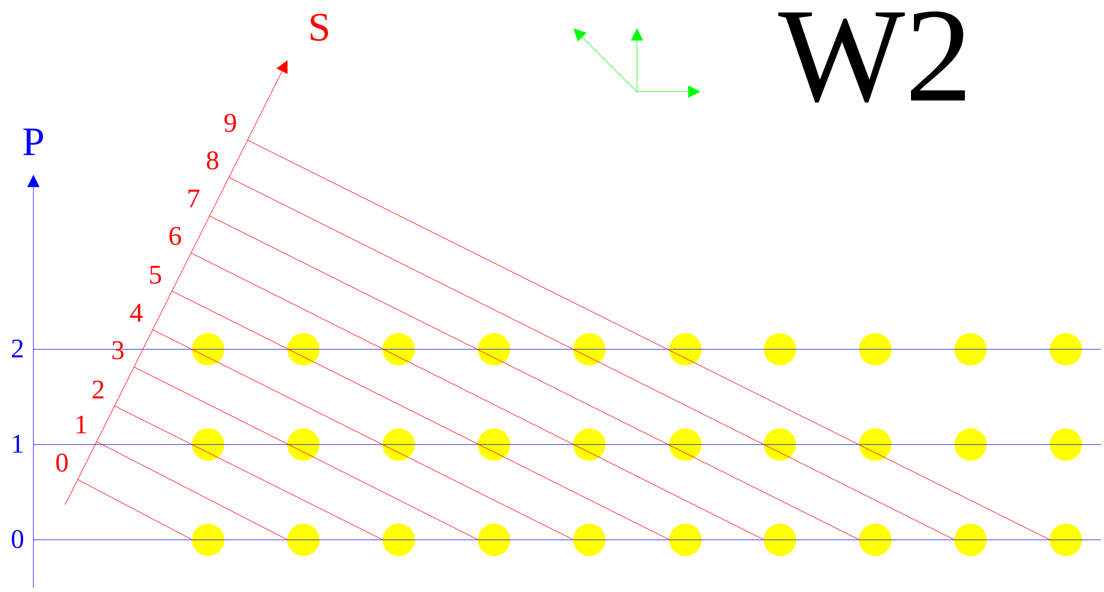
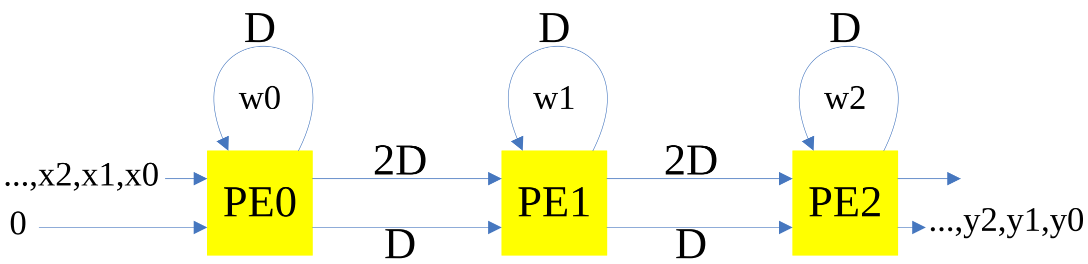


图 12　W2 脉动阵列映射图（权重保持，输入和结果同方向但不同速度移动）

如图 12，选择 $P=\begin{bmatrix}0\\1\end{bmatrix}$ 及 $S=\begin{bmatrix}1\\2\end{bmatrix}$，显然 P 和 S 不平行，满足“可行性限制条件”。显然 W2 需设置 3 个 PE，并且需反转输出边。边的映射结果表如下

边	P^Te	S^Te
权值 $[1,0]^T$	0	1
输入 $[0,1]^T$	1	2
输出 $[-1,1]^T$	1	1

脉动结构为



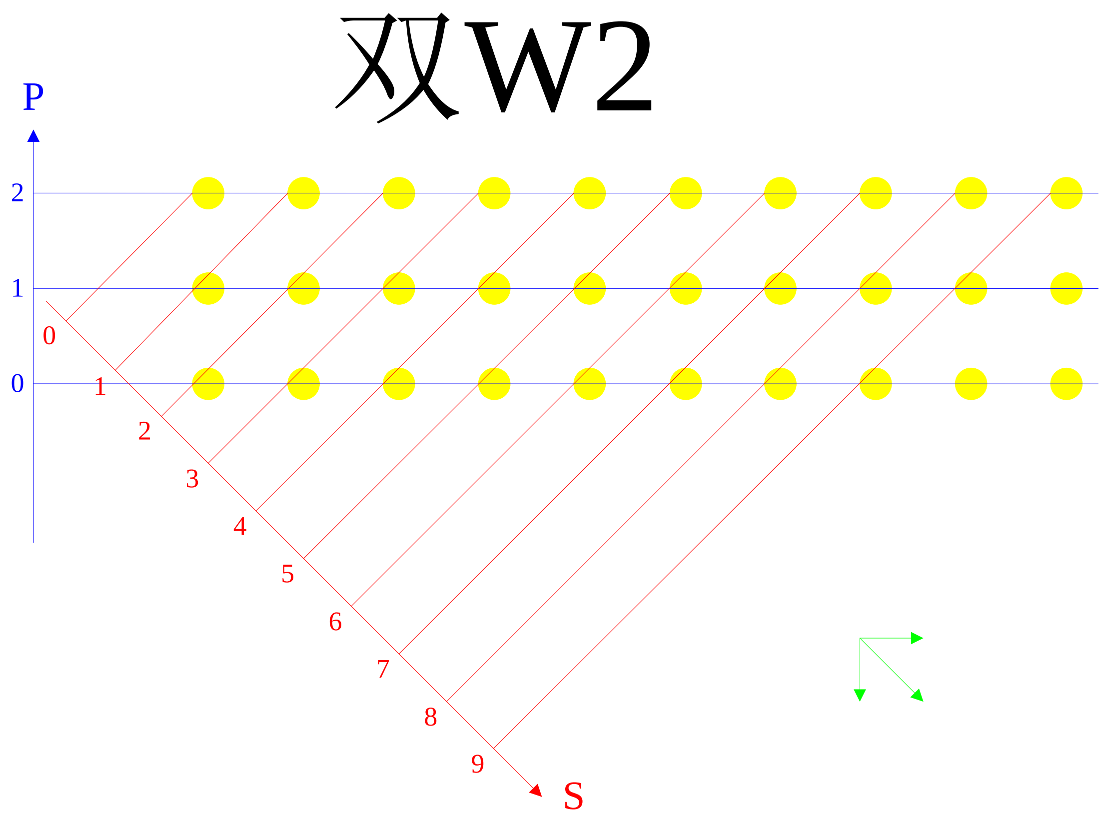
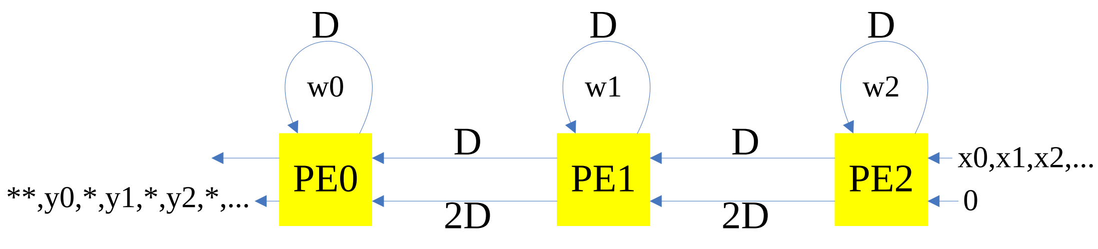


图 13　双 W2 脉动阵列映射图（权重保持，输入和结果同方向但不同速度移动）

如图 13，选择 $P=\begin{bmatrix}0\\1\end{bmatrix}$ 及 $S=\begin{bmatrix}1\\-1\end{bmatrix}$，显然 P 和 S 不平行，满足“可行性限制条件”。显然双 W2 需设置 3 个 PE，并且需反转输入边。边的映射结果表如下

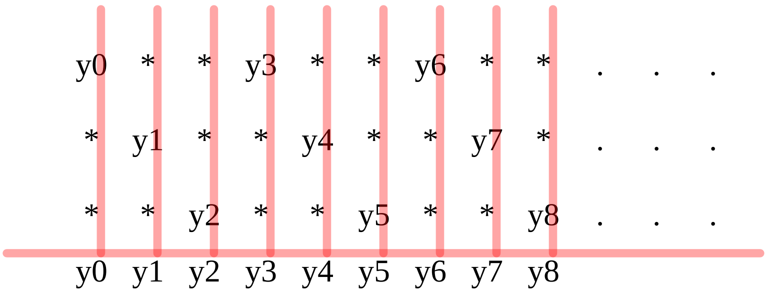
边	$P^T e$	$S^T e$
权值 $[1,0]^T$	0	1
输入 $[0,-1]^T$	-1	1
输出 $[1,-1]^T$	-1	2

脉动结构为



一口气看完那么多种 3 阶 FIR 滤波器的脉动结构，大家应该对什么是脉动，怎么来设计脉动结构，以及什么叫投影设计法有比较清楚的了解了吧。这一章的内容的确是比较有挑战性，对大脑是极好的锻炼！！大家要奋发，切不可半途而废。

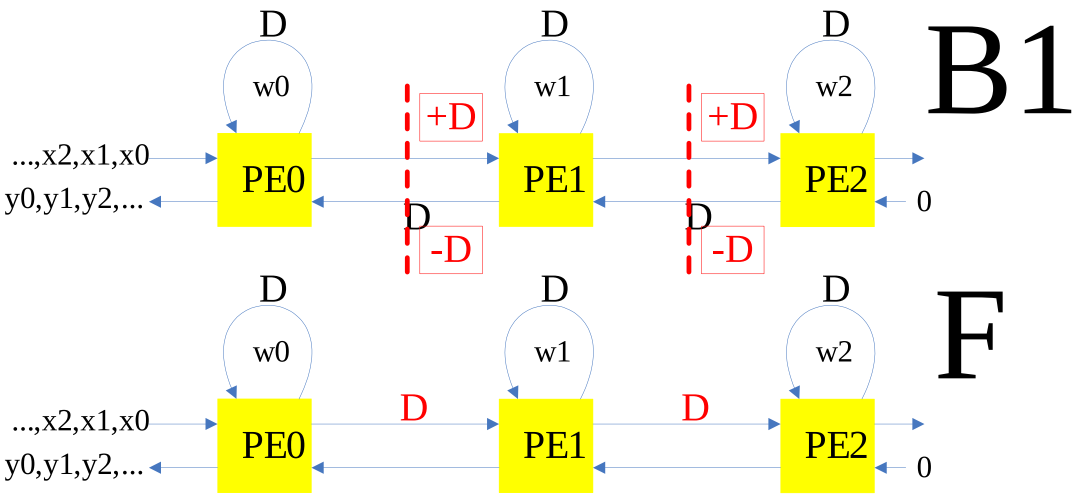
很多同学看完前面的脉动结构设计后，不明白权值以及输入输出序列到底是如何安排的？比如双 W2 的结构，输入序列按“x0,x1,x2,…”从高序号 PE 传送到低序号 PE，这表示 0 周期 x0 到达 PE2；1 周期 x1 到达 PE2 而 x0 到达 PE1；2 周期 x2 到达 PE2，x1 到达 PE1，同时 x0 到达 PE0，等等以此类推。输出序列按“**,y0,*,y1,*,y2,*,…”，这里一个*表示一个无效数据，当然了**表示接连的两个周期都输出无效数据，该输出序列的意义是在 0 和 1 两个周期输出*数据，在 2 周期输出 y0，3 周期又是一个*，4 周期是 y1，也就是从 y0 开始，每个一个周期输出一个有意义的 y。再比如双 R2 结构，每个处理器在特定周期都有输出，请看下图，一目了然，



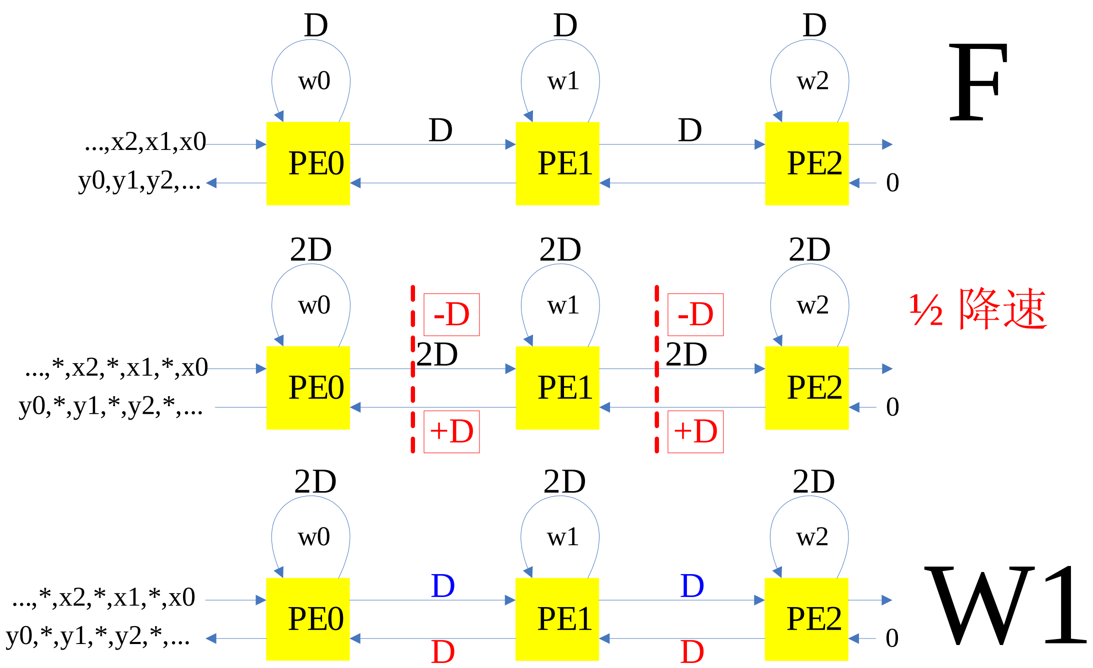
从上到下，第一行是 PE0 的输出，第二行是 PE1 的输出，第三行是 PE2 的输出，这些输出组合起来，不就是每个周期输出一个 y；因为 PE 是轮流输出，所以对一个 PE，必须每隔两个周期才会得出一个有效结果。

设定不同的 P 和 S 轴，将会构造出不同形式的脉动阵列，但是这些阵列是“相通”的，且功能相同。所谓的“相通”是指可以通过各种电路变化从某一个阵列导出另一个阵列，课本上给出了例子，比如 F 可以通过 B1 应用割集重定时得到，W1 可由 F 二分之一减速然后在应用割集重定时得到。

示例 1，



示例 2，

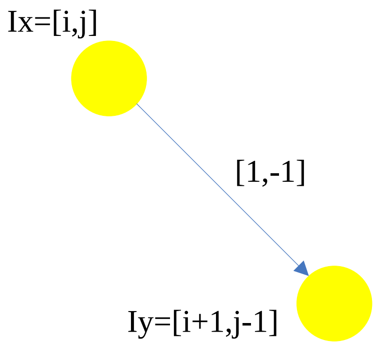


前面所构造的脉动空间隐含了很大的人为因素，而且还默认那么一点：驱动系统的时钟周期 T 足够长，以至于不用考虑节点关键路径的长短问题。比如 B1 结构，关键路径只集中在节点内部，所以 T 只需大于等于节点自身的执行时间即可；但对于 F 结构，关键路径同时由[1,-1]方向的三个节点同时决定，此时 T 就要大于单个节点执行时间，大多少由具体情况决定。在 FIR 的例子中节点包含一个乘法单元和加法单元，假设乘法单元计算时间为 Tm，加法单元计算时间为 Ta，那么对 B1 结构要求 T>=Tm+Ta 即可，对 F 要求 T>=Tm+2Ta（也许你认为应该是 T>=Tm+3Ta，那也没错，只是边界节点的加法器可以去掉）。

在实际的系统设计中，如果 T 是预先规定的且不能更改，而节点的计算时间偏偏又大于 T，此时就不能“随心所欲”地设定 S 和 P 矢量。特别是 S，S 选择不当将导致节点计算结果错误。**怎么在这种约束下，确定 S 矢量呢？**

同样以图 1 的 DG 为例，假设节点需要 2T 才能计算完毕。直观的想，每个节点应该在 PE 停留 2 个周期，怎么选择 S 调度矢量才能实现这一点？

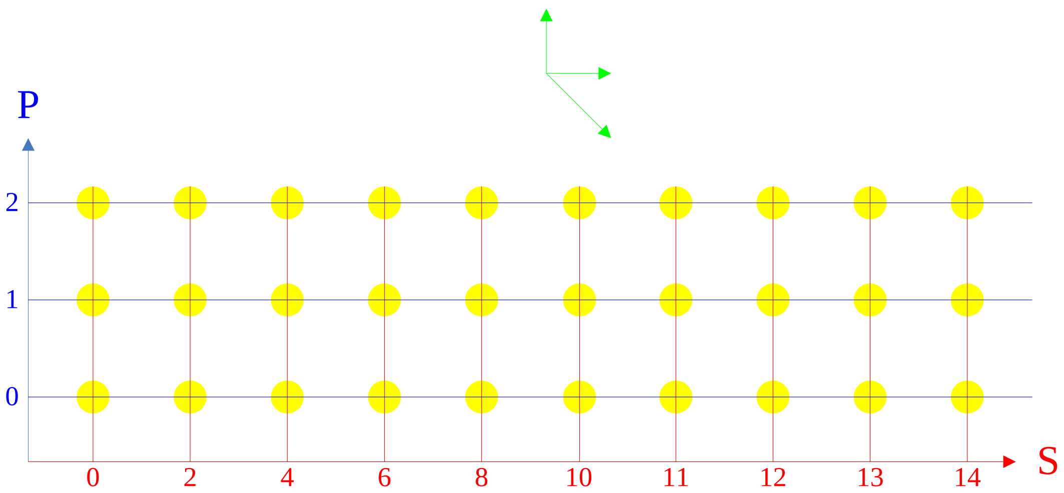
解答：由图 1 的 DG 可知，节点之间真正存在关系的边是[1,-1]的输出边。考虑如下两个节

点

。设 $S=\begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$，则 Ix 在 $S \cdot I_x = s_1 \times i + s_2 \times j$ 周期被调度，Iy 在 $S \cdot I_y = s_1 \times (i+1) + s_2 \times (j-1)$ 周期被调度，那么使得 $S \cdot I_y \geq S \cdot I_x + 2$ 成立的 S 就是所求解。化简不等式，有

$$s_1 - s_2 \geq 2$$

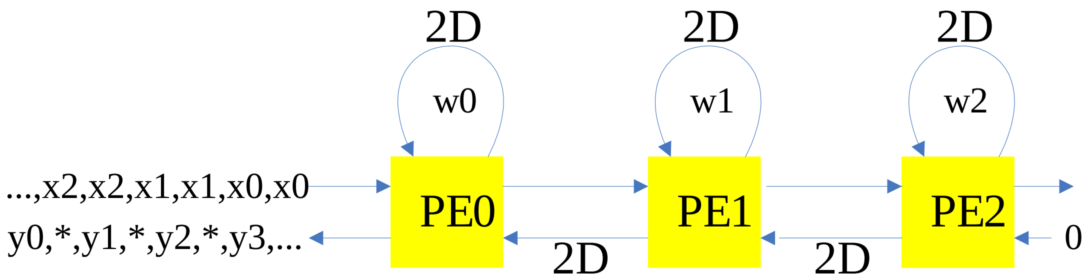
取 $S=\begin{bmatrix} 2 \\ 0 \end{bmatrix}$，确定 S 之后只需保证 P 不与 S 平行即可，不妨取 $P=\begin{bmatrix} 0 \\ 1 \end{bmatrix}$，所得投影图如下，



边映射关系为

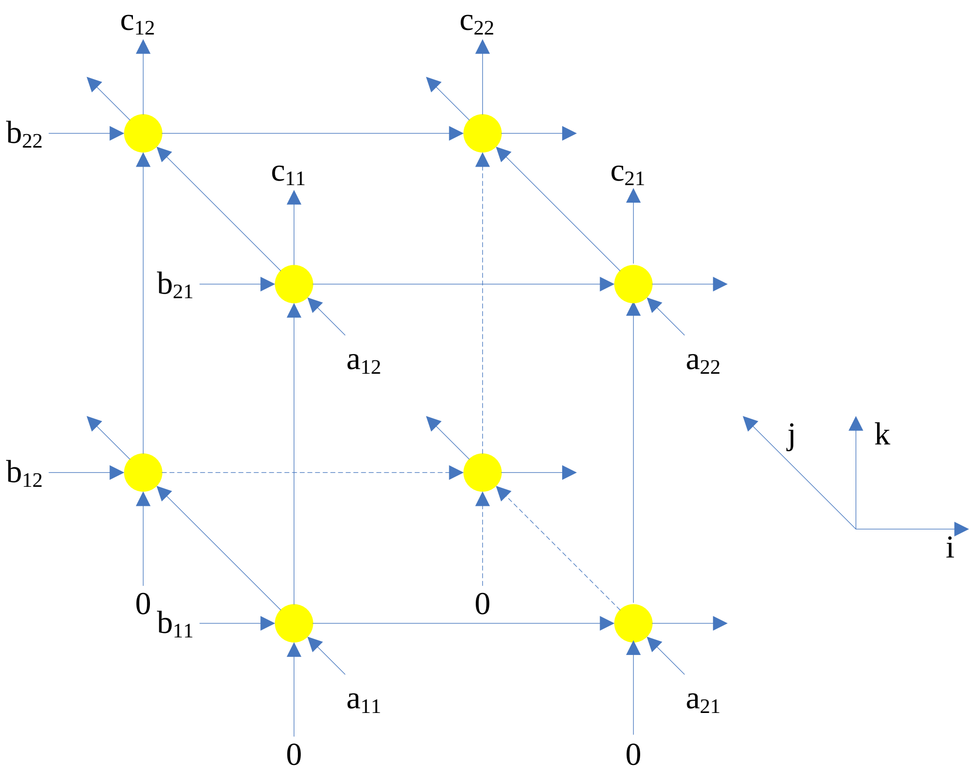
边	P^Te	S^Te
权值 $[1,0]^T$	0	2
输入 $[0,1]^T$	1	0
输出 $[1,-1]^T$	-1	2

脉动结构如下



课本 7.4 节给出较为详细的描述，并给出了 RIA 和 RDG 的概念。因为这些内容并不是很难，而且前面的展开和折叠我们一直在和“调度”打交道，7.4 节的内容其实就是建立调度不等式，并求解得出合理的调度矢量 S，留给大家做练习！

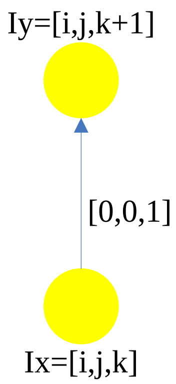
接下来，我们做些更有挑战的设计。如图 2 所示 2x2 矩阵乘法 DG，如何来设计相应的脉动结构呢？



不同于 FIR 的例子，矩阵乘法是 3 维 DG，也许可用投影法将其投影到 2 维脉动空间。

注意，2 维脉动空间是这样一个 3 维空间，其中 2 个维度是 PE 空间，也就是 PE 构成平面网络，另一个维度是时间。

在图 2 的 DG 中，每个节点的内容是一个乘法器和一个加法器。在这个 DG 中，调度的约束在于[0,0,1]边，这条边表示将前一个节点的结果和当前节点所得的 a*b 相加。如下图示



，令 $S=\begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}$，则调度不等式为 $S\cdot I_y \ge S\cdot I_x+1$，化简得

$$s_3 \ge 1$$

注意，课本上对所有边建立调度不等式是严格正确的。这里我之所以只对$[0,0,1]^T$ 边建立调度不等式，是因为其他边其实是一种“广播”性质的边，不会对导出正确的脉动结构造成影响。为了验证这一点，不妨取 $S^T=\begin{bmatrix}-1 & -1 & 1\end{bmatrix}$，$P^T=\begin{bmatrix}1 & 0 & 0 \\ 0 & 1 & 0\end{bmatrix}$，这个解按课本上的做法是非法解，其实不然，请往下看。这里需要进行 a 和 b 边的反转，才能保证映射是合法的，边映射如下表

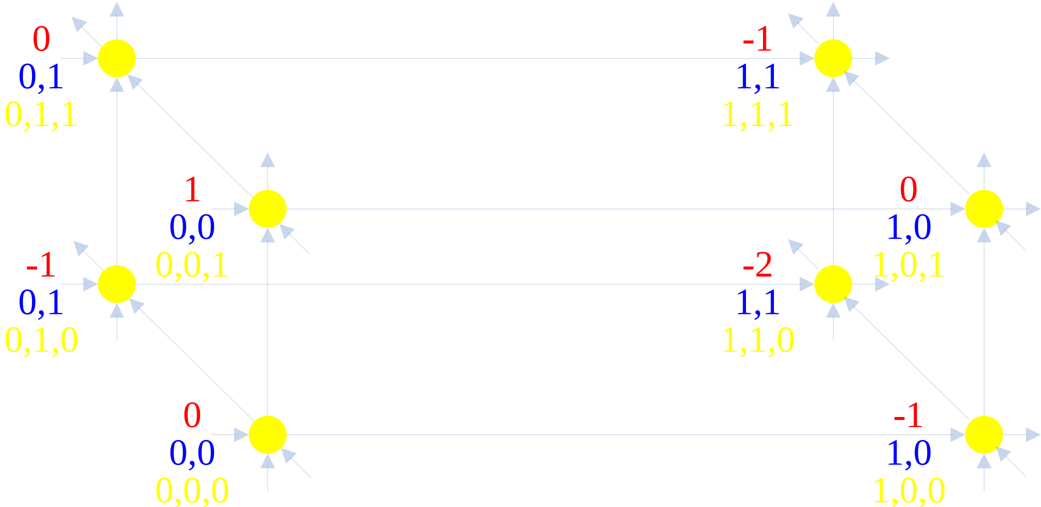
边	P^Te	S^Te
a $[0,-1,0]^T$	$[0,-1]^T$	1
b $[-1,0,0]^T$	$[-1,0]^T$	1
c $[0,0,1]^T$	$[0,0]^T$	1

注：$P^T=\begin{bmatrix}1 & 0 & 0 \\ 0 & 1 & 0\end{bmatrix}$表示 PE 平面，$[1,0,0]^T$和$[1,0,0]^T$构成该平面的两个坐标轴，在选择 S 和 P 时必须考虑“可行性限制条件”。判断 S 是否平行于 P，等价于判断 S 是否和 P 的两个坐标基的叉乘正交：如果 S 和 P 的基矢量叉积正交，S 平行 P，反之则不然。对于以上的 S 和 P 有

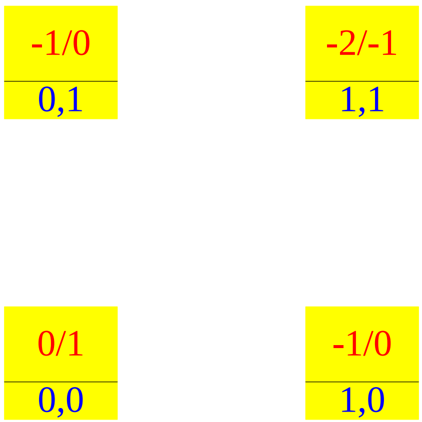
$$\left(\begin{bmatrix}1 & 0 & 0\end{bmatrix}\otimes\begin{bmatrix}0 & 1 & 0\end{bmatrix}\right)\cdot\begin{bmatrix}-1 & -1 & 1\end{bmatrix}=1\neq 0$$

所以 S 不平行与 P。

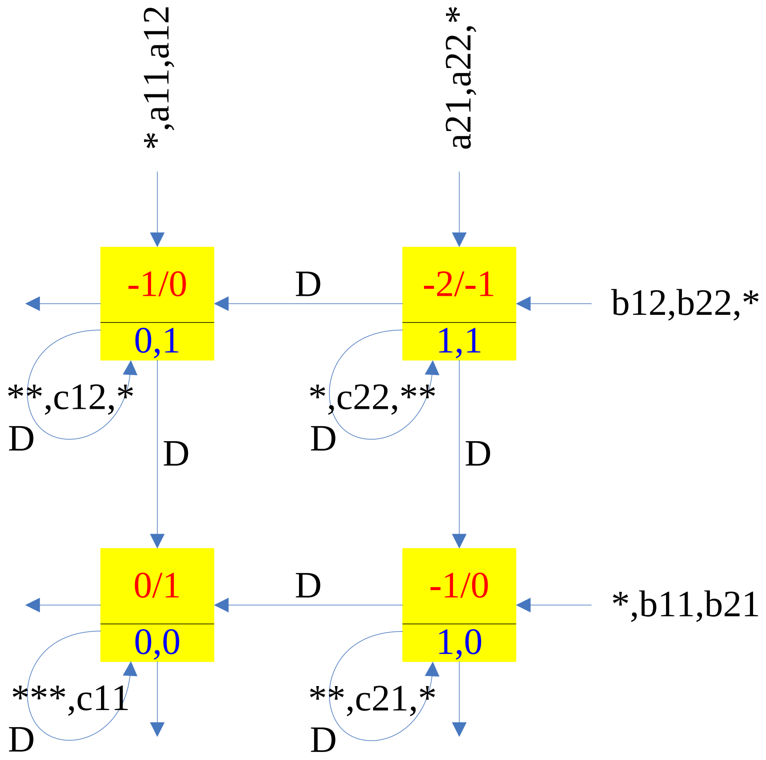
不像 FIR 的例子那么简单，可以形象的画出投影的情况。在这个例子中只能重点依赖于数字而不是图形了。首先把 DG 中各个节点的调度时间和所分配的 PE 序号标出来



图中，黄色数字是节点的坐标，蓝色数字是节点映射的 PE 序号，红色数字是节点被调度的时刻（相对时间）。根据这个图，就很容易画出脉动结构了，从图中可知，可能出现的 PE 序号有(0,0)/(0,1)/(1,0)/(1,1)，所以只需在处理器平面设置 4 个 PE 节点即可，如图



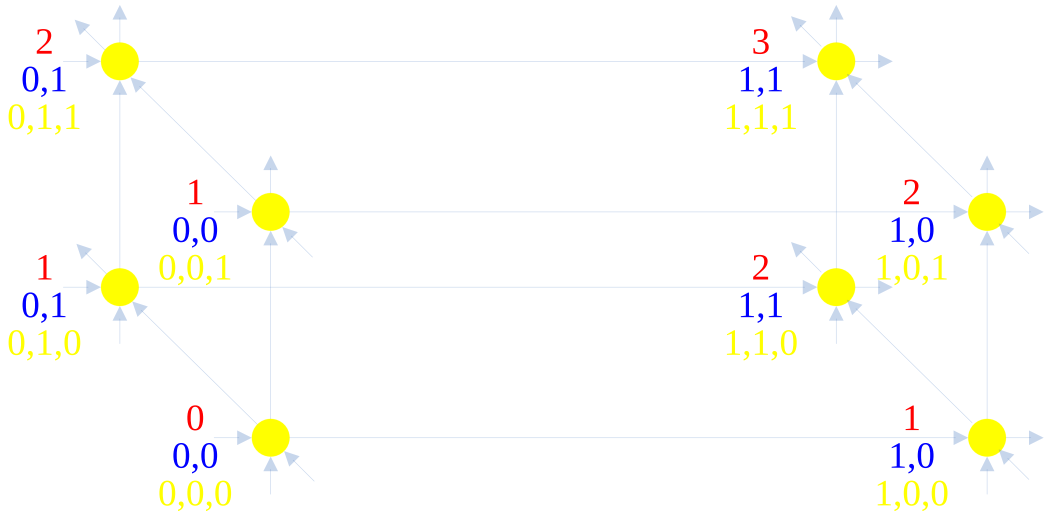
上图给出了每个 PE 的序号，以及该 PE 工作的时刻，比如左下角为(0,0)号 PE，在 0 周期和 1 周期工作，同理右上角为(1,1)号 PE，在-2 周期和-1 周期工作。结合上面的两个图，可以清楚的知道哪一个节点在哪一个周期被调度到哪一个 PE 运行。接下来将边映射到脉动阵列中，有



大家可以验证一样功能是否正确。

接下来，逐个构造课本上所列出的 7 个解。

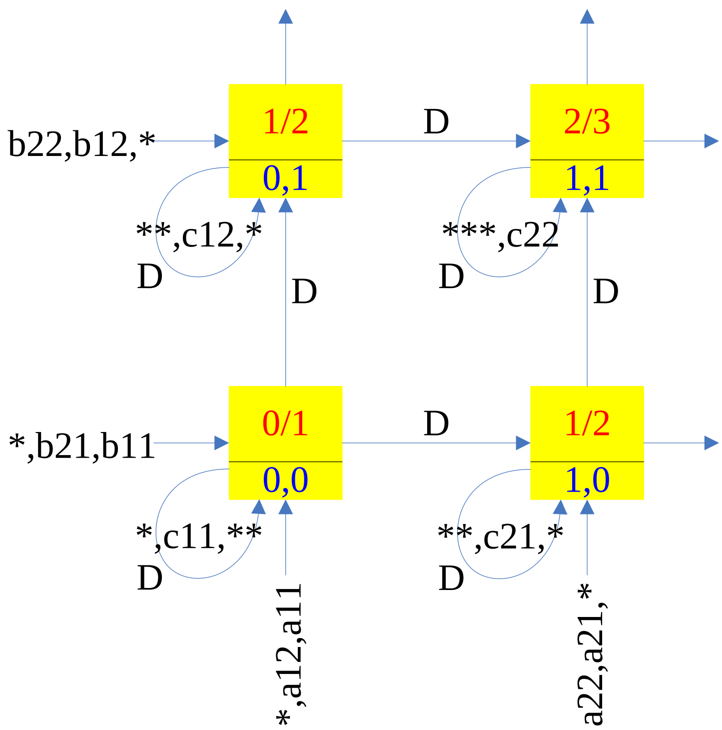
解 1： 取 $S^T=\begin{bmatrix}1 & 1 & 1\end{bmatrix}$，$P^T=\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\end{bmatrix}$。调度图如下



边映射表

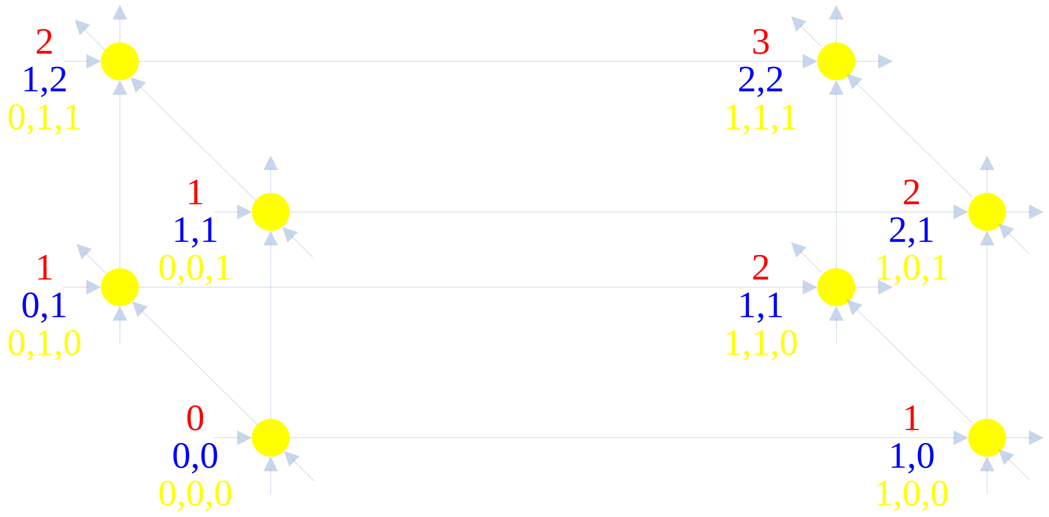
边	P^Te	S^Te
a $[0,1,0]^T$	$[0,1]^T$	1
b $[1,0,0]^T$	$[1,0]^T$	1
c $[0,0,1]^T$	$[0,0]^T$	1

脉动结构如下



解 2：

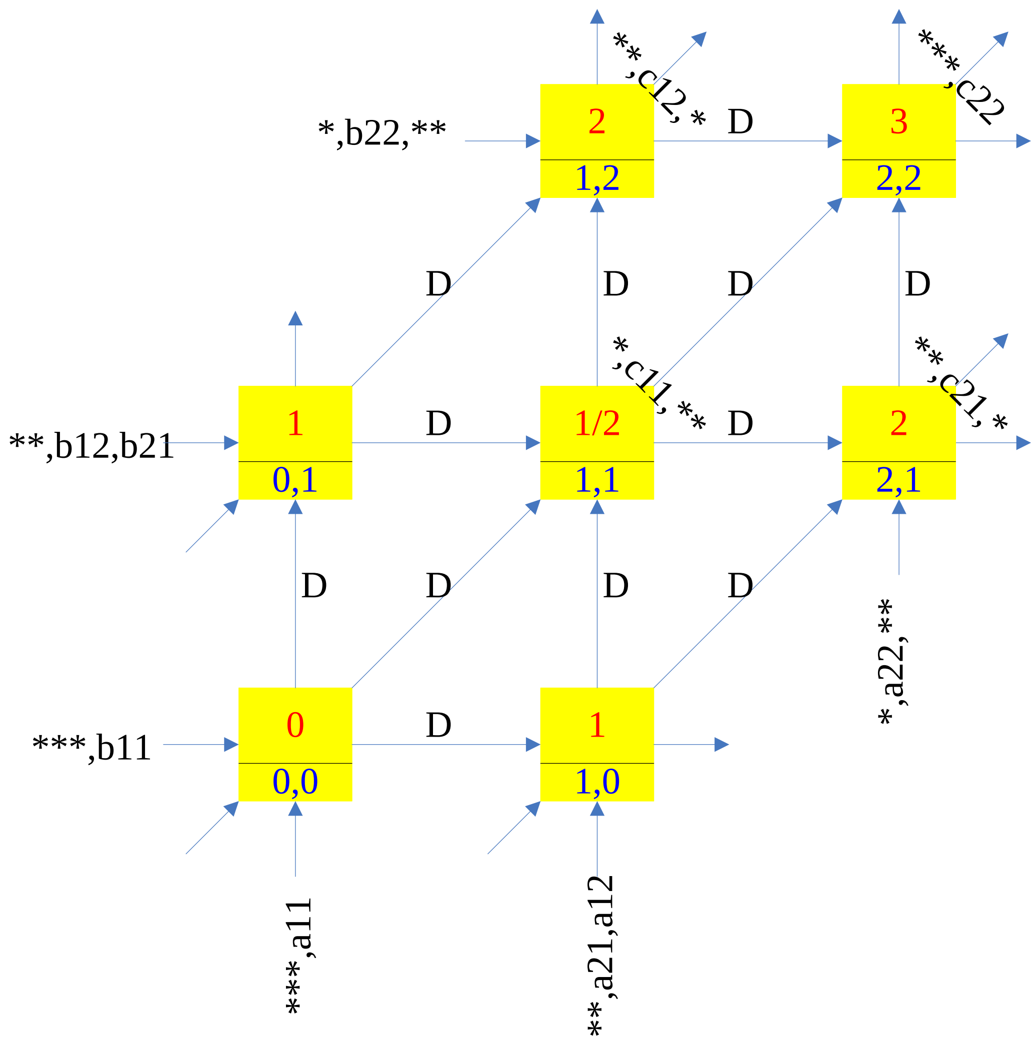
取 $S^T=[1\quad 1\quad 1]$，$P^T=\begin{bmatrix}1&0&1\\0&1&1\end{bmatrix}$。调度图如下



边映射表

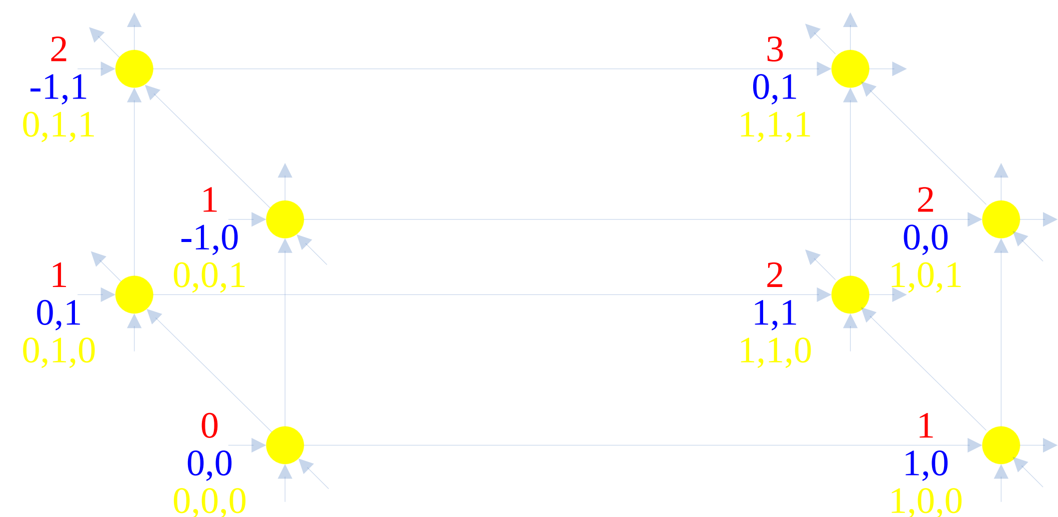
边	$P^T e$	$S^T e$
a $[0,1,0]^T$	$[0,1]^T$	1
b $[1,0,0]^T$	$[1,0]^T$	1
c $[0,0,1]^T$	$[1,1]^T$	1

脉动结构如下



解 3:

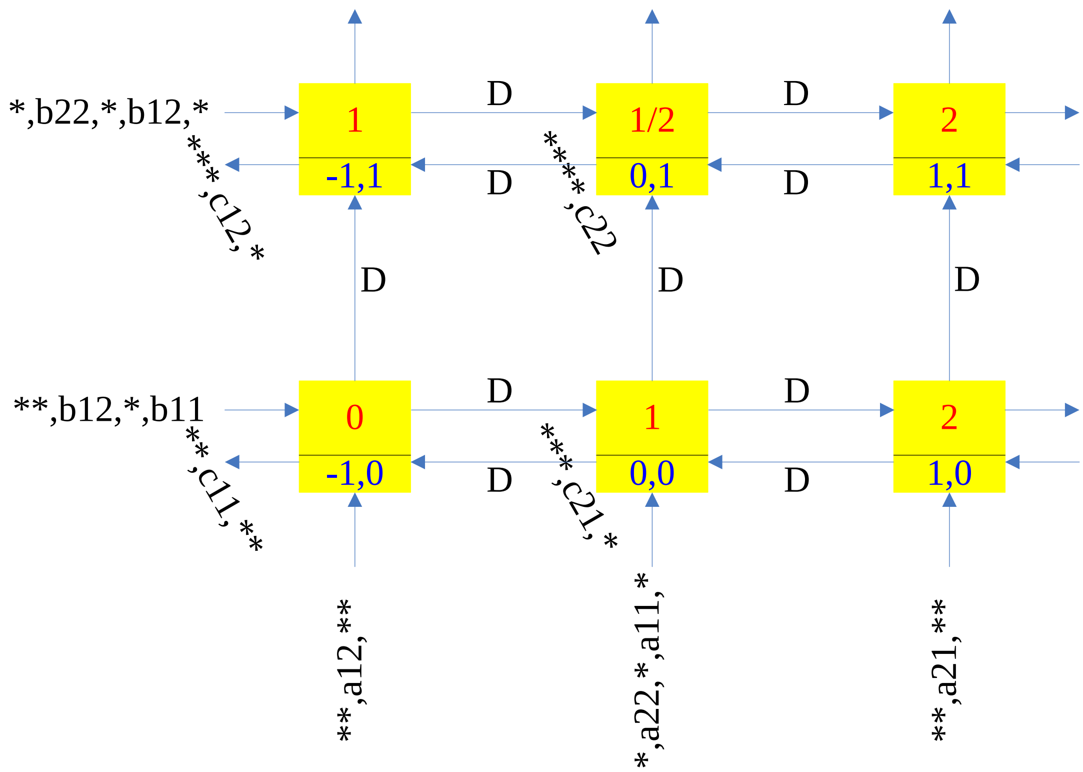
取 $S^T = [1 \quad 1 \quad 1]$，$P^T = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$。调度图如下



边映射表

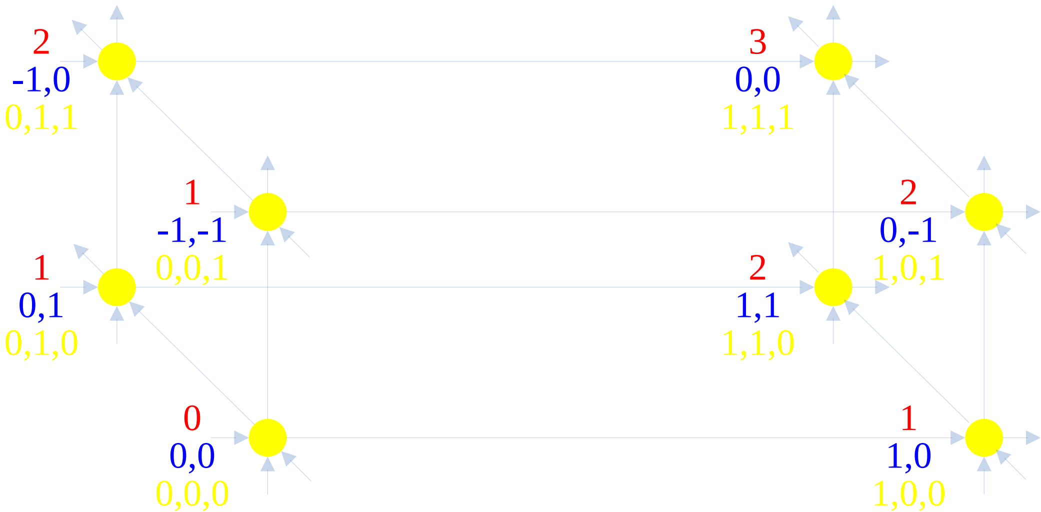
边	$P^T e$	$S^T e$
a $[0,1,0]^T$	$[0,1]^T$	1
b $[1,0,0]^T$	$[1,0]^T$	1
c $[0,0,1]^T$	$[-1,0]^T$	1

脉动结构如下



解 4:

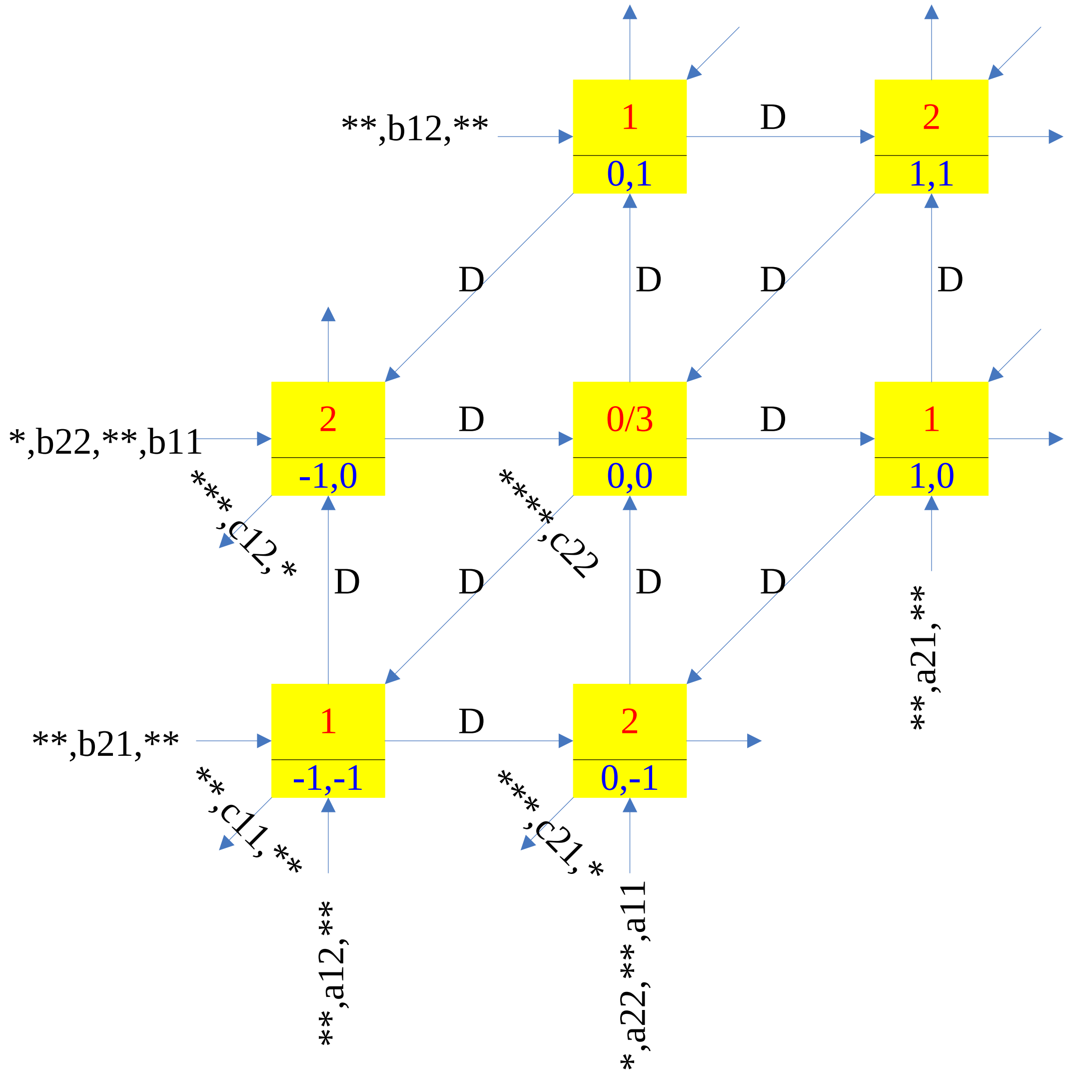
取 $S^T = [1 \quad 1 \quad 1]$，$P^T = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}$。调度图如下



边映射表

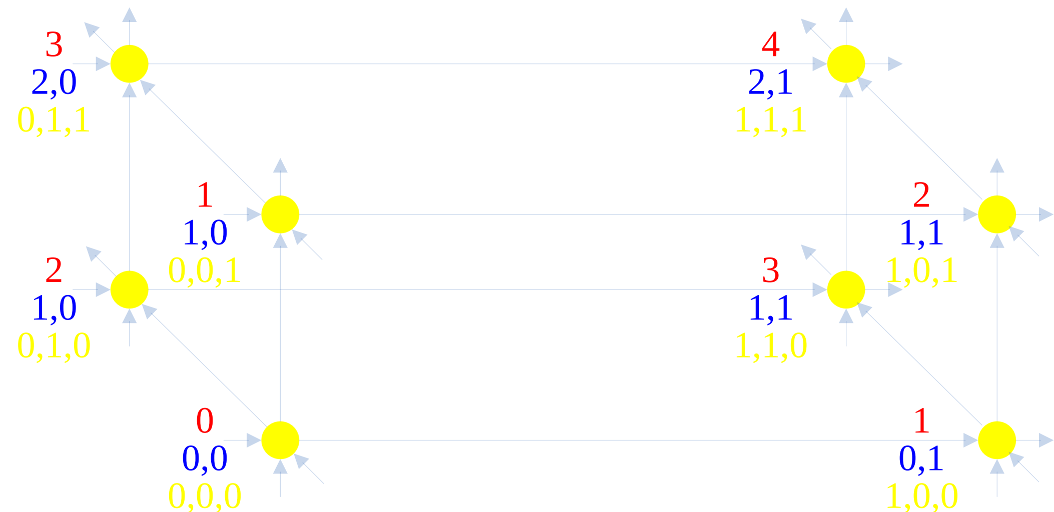
边	$P^T e$	$S^T e$
a $[0,1,0]^T$	$[0,1]^T$	1
b $[1,0,0]^T$	$[1,0]^T$	1
c $[0,0,1]^T$	$[-1,-1]^T$	1

脉动结构如下



解 5:

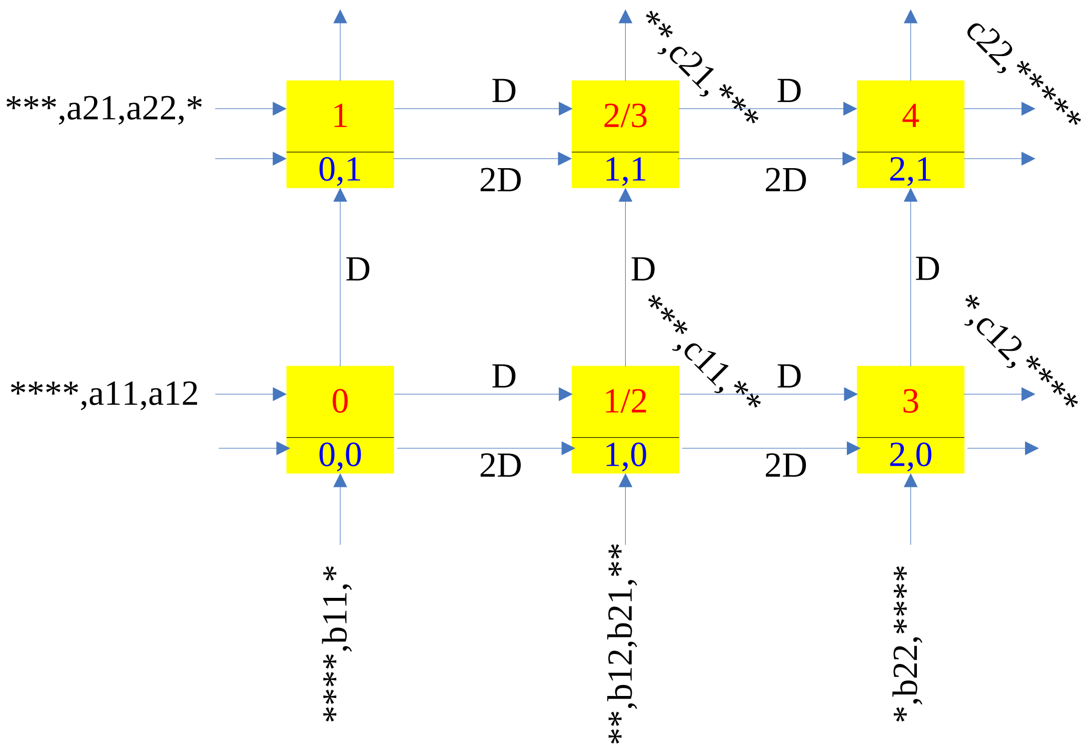
取 $S^T=\begin{bmatrix}1 & 2 & 1\end{bmatrix}$，$P^T=\begin{bmatrix}0 & 1 & 1\\ 1 & 0 & 0\end{bmatrix}$。调度图如下



边映射表

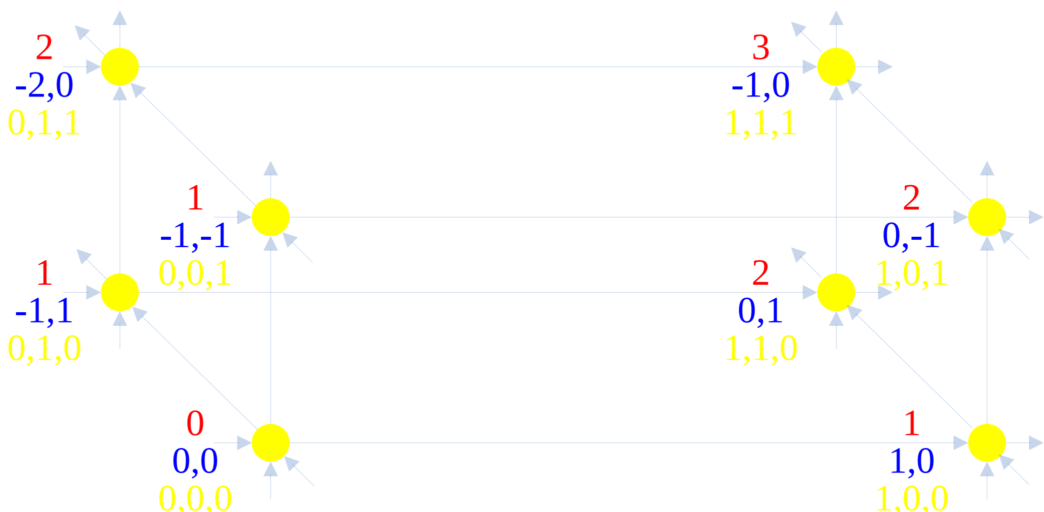
边	$P^T e$	$S^T e$
a $[0,1,0]^T$	$[1,0]^T$	2
b $[1,0,0]^T$	$[0,1]^T$	1
c $[0,0,1]^T$	$[1,0]^T$	1

脉动结构如下



解 6:

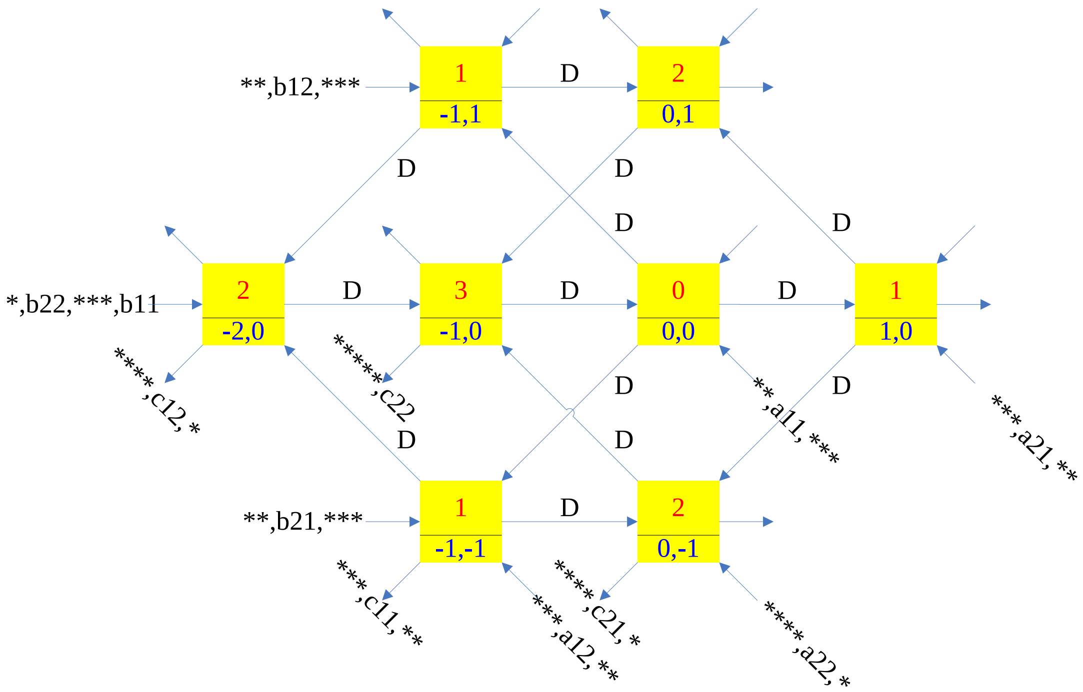
取 $S^T = \begin{bmatrix}1 & 1 & 1\end{bmatrix}$，$P^T = \begin{bmatrix}1 & -1 & -1\\ 0 & 1 & -1\end{bmatrix}$。调度图如下



边映射表

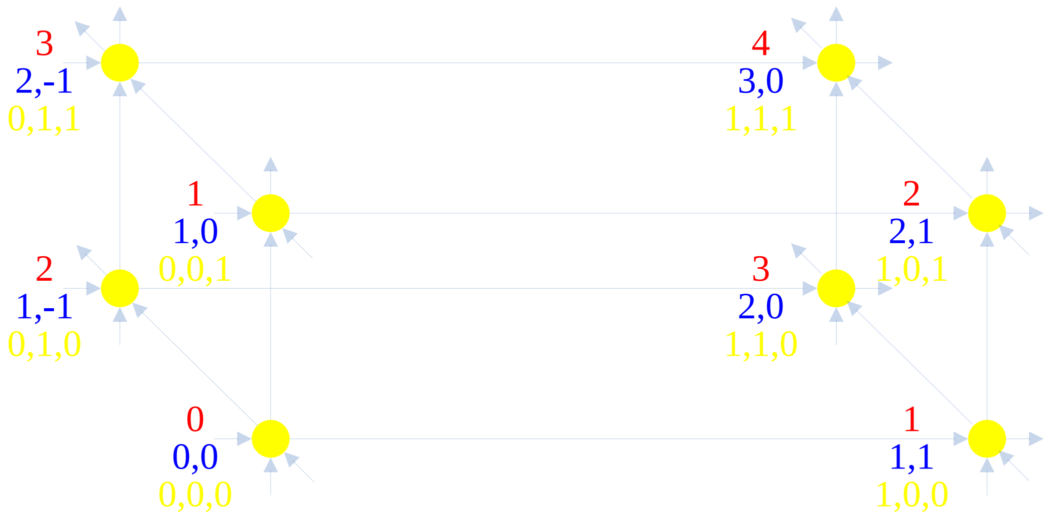
边	$P^T e$	$S^T e$
a $[0,1,0]^T$	$[-1,1]^T$	1
b $[1,0,0]^T$	$[1,0]^T$	1
c $[0,0,1]^T$	$[-1,-1]^T$	1

脉动结构如下



解 7:

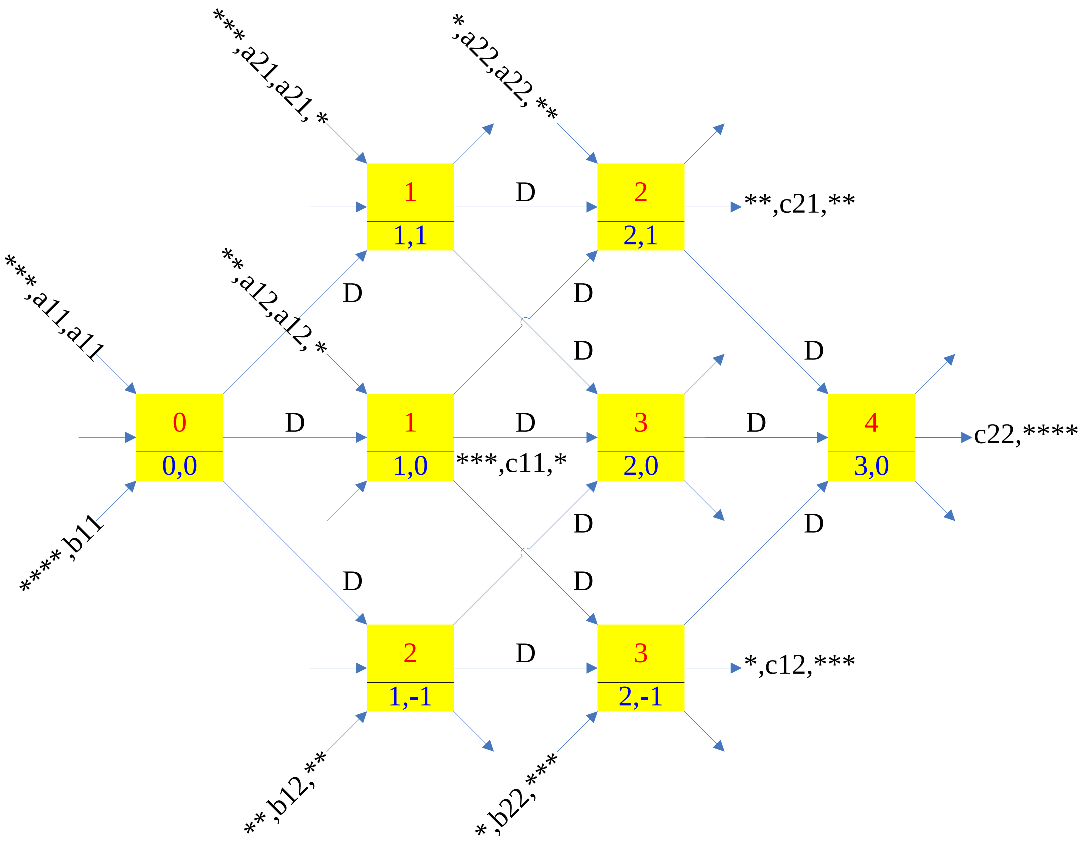
取 $S^T=\begin{bmatrix}1 & 2 & 1\end{bmatrix}$，$P^T=\begin{bmatrix}1 & 1 & 1\\ 1 & -1 & 0\end{bmatrix}$。调度图如下



边映射表

边	$P^T e$	$S^T e$
a $[0,1,0]^T$	$[1,-1]^T$	2
b $[1,0,0]^T$	$[1,1]^T$	1
c $[0,0,1]^T$	$[1,0]^T$	1

脉动结构如下



对于 2x2 矩阵乘法，课本上给出的解并不是硬件利用率最高的解，有兴趣的同学可以试试这个，$S^T=[0,0,1]$ 且 $P^T=\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\end{bmatrix}$。此外很多解的 PE 资源其实可以折叠（复用），比如解 1，你能构造出只用 3 个 PE 的解 1 脉动结构吗（提示，在 PE 序号上做手脚）？

小结：脉动设计其实就是将 DG 按某种规则投影到脉动空间，注意，脉动空间有一个维度是时间，其他维度构成脉动网络。冥冥的貌似脉动和折叠有某种本质联系，都是将节点的任务分配到具体处理单元，并规划好处理单元之间的互联关系。也许随着学习的深入，我们会悟出一些本质的东西，进而把所有的设计方法统一起来。

本章的内容比较有挑战性，所以错误真的是在所难免，大家应该相信自己，如果你觉得某些地方有问题，尽管提出来，以便我改正。